all rights reversed

HACKER

Απρίλης 2013 # Τιμή τεύχους 6,40€

τεύχος 019 - Northern Lights Edition

ΑΝηλεής [*] ΑΝάλυση ΠΑΚΕΤΩΝ

Ήχοι από το παρελθόν

Το Dropbox του φτωχού

Σπιτικές βιβλιοθήκες για το Arduino

[*] ...επειδή έχει τρομερό ενδιαφέρον αλλά και γιατί μπορούμε

tefhos 19o

Περιεχόμενα

και οδηγίες χρήσης

Εκδότης / Διευθυντής	Χρήστος Βαρελάς
Αρχισυντάκτης	Παναγιώτης Βαρελάς
Σύμβουλοι έκδοσης	Νίκος Μουρατίδης, Βούλα Παυλίδου
Σύνταξη	Παναγιώτης Βαρελάς, Χρήστος Βαρελάς, Ανδρέας Βενιέρης, Γιώργος Γιάννου, Μανώλης Κιαγιάς, Ιωάννης Κονιάρης, Γιώργος Μελέκος, Νίκος Μουρατίδης, Νικόλαος Σηφάκης, Μέσος Παπαδόπουλος (TM), Χρίστος Τόμπρας
Γραμματική επιμέλεια	Βούλα Παυλίδου
Καλλιτεχνική Επιμέλεια	Πέτρος Φιλιππίδης

Το περιοδικό deltaHacker είναι συνδρομητικό και εκδίδεται από την Parabing Creations δώδεκα φορές το χρόνο, κάθε μήνα. Η Parabing Creations έχει την έδρα της στη Μαιώτιδος 3, 55 133, Καλαμαριά. Η εκτύπωση και η βιβλιοδεσία γίνονται από την ThessPrint A.E. Το email επικοινωνίας του περιοδικού είναι το talk2us@deltahacker.gr και το email για τις συνδρομές το subscriptions@deltahacker.gr. Οι πληροφορίες για τις συνδρομές είναι στο http://deltahacker.gr/subscriptions και οι παραγγελίες γίνονται από το http://deltahacker.gr/order. Όλες οι απόψεις που εκφράζονται στα άρθρα δεν εκφράζουν απαραίτητα και τη γνώμη του περιοδικού.

(*) Πότε θα ξεκινήσετε τη δημιουργία του πιο ζηλευτού box;
http://deltahacker.gr/deltacast-s01e08



Πλήρη **άρθρα που συμπληρώνουν** το περιοδικό. Το **video podcasts** του περιοδικού [deltaCast]. Τα προκλητικά **mind hacks**. Όλα τα βίντεο από τα **live streams** [deltaStream]. Ολόκληρα **blasts from the past**. Τα πάντα ενδιαφέροντα **fora**. Το σημείο συνάντησης για τους φίλους του περιοδικού.

deltahacker.gr

Αληθινό, διασκεδαστικό hacking **για όλους!**

Επιθυμία Vs Πραγματικότητα

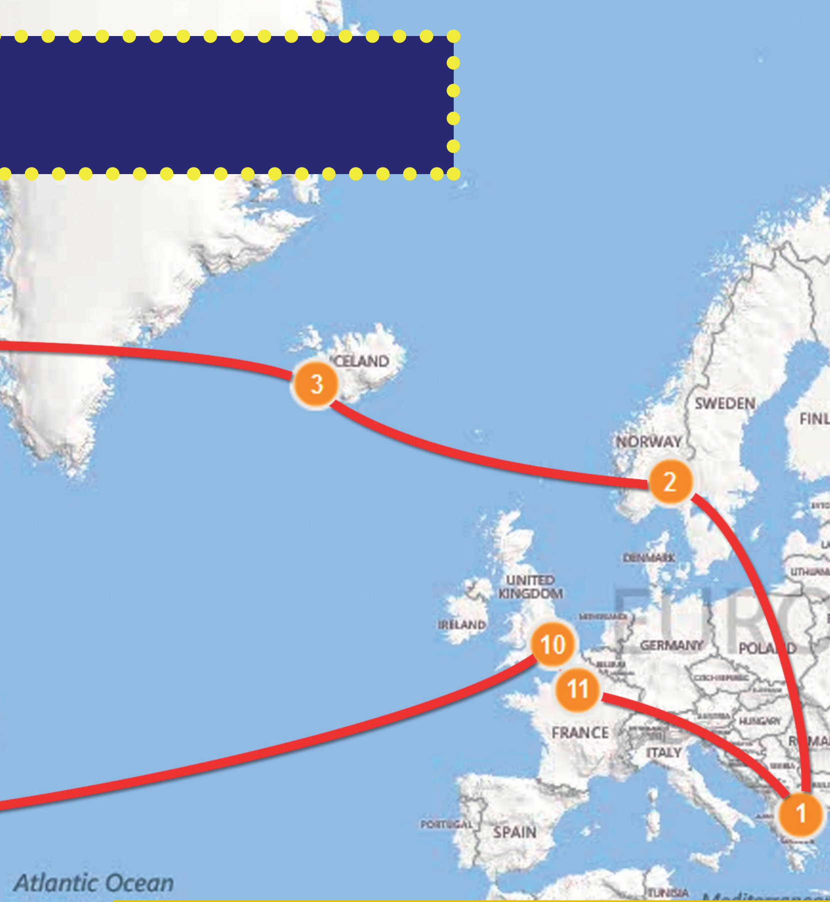
Πριν ξεκινήσω τη συγγραφή του editorial σκόπευα να βάλω ως τίτλο κάτι σαν «Μου αρέσει Vs Πρέπει». Λίγο αργότερα που το σκέφτηκα καλύτερα συνειδητοποίησα ότι, κατά πώς φαίνεται, υπάρχουν νοηματικά μονοπάτια που οδηγούν από τον αρχικό τίτλο στον άλλον, αυτόν δηλαδή που τελικά επέλεξα. Μετά λοιπόν τη διευθέτηση του ζητήματος «τίτλος», που για κάποιους από εμάς είναι πάντα κρίσιμο, άρχισα επιτέλους να γράφω έχοντας μια γενική ιδέα για το τι περίπου θέλω να πω, αλλά χωρίς να είμαι σίγουρος για το πού θέλω να καταλήξω. Ευτυχώς που όλοι λένε ότι το ταξίδι έχει σημασία κι όχι ο προορισμός, οπότε όλα καλά...

by subZraw

Πάντως η κουβέντα που είχα τις προάλλες με τη Σύμβουλο Εκδόσεως (TM) του περιοδικού, έι-κέι-έι @citr0nella, είχε να κάνει κυρίως με προορισμούς. Στο πλαίσιο ενός αναπάντεχου session που αποτελούταν κατά 87% από το λεγόμενο day-dreaming, χωρίς καλά καλά να το συνειδητοποιήσουμε αρχίσαμε να σχεδιάζουμε το τέλειο δρομολόγιο για τις τέλειες διακοπές, τις οποίες κάποια στιγμή οπωσδήποτε θα κάνουμε — μονοκοπανιά ή σε (πολλές) δόσεις.

Το δρομολόγιο αρχίζει με πτήση από Θεσσαλονίκη προς Όσλο. Μετά από λίγες μέρες εκεί έχει Ρέικιαβικ (ούτε καν τρίωρη πτήση) και ύστερα από μια βδομάδα φεύγουμε για Άνκορατζ. Ακολουθούν Σιάτλ, Σαν Φρανσίσκο και Λος Άντζελες. Κάπου εδώ, η Σύμβουλος Εκδόσεως (TM) προσπάθησε να προπαγανδίσει υπέρ νοτίου Αμερικής και συγκεκριμένα υπέρ Αργεντινής. Περιέργως η Γη του Πυρός δεν την ενδιέφερε, ούτε εκείνο το παγοθραυστικό που κάνει κρουαζιέρες προς Ανταρκτική. Έτσι, μετά από ομοβροντία λογικών επιχειρημάτων τα οποία κατά κύριο λόγο περιστρέφονταν γύρω από τον άξονα του «δεν θέλω», κατάφερα να την πείσω για βορειανατολική πορεία και συγκεκριμένα προς Ντένβερ (μη ρωτάτε) και μετά Νέα Υόρκη (κλασικά). Αφήνοντας την αμερικανική ήπειρο —και σημειώνοντας εμφατικά ότι φυσικά δεν ξεχάσαμε τον Καναδά, απλά τον αφήνουμε για άλλη φορά— πετάμε προς Λονδίνο και περιέργως αισθανόμαστε ότι επιστρέφουμε στη βάση μας. Καμία σχέση και οι δύο με Λονδίνο αλλά, να, τόσες βδομάδες across the pond, με το που επιστρέφουμε Ευρώπη κάτι παθαίνουμε. Μετά από μερικές μέρες χαλάρωσης στο Λονδίνο σειρά έχει ίσως ένα Παρίσι και ύστερα απ' αυτό επιστροφή στη Θεσσαλονίκη.

Η Θεσσαλονίκη, ως γνωστόν, είναι η Μεγάλη Φτωχομάνα (TM), κι αυτή η συνειδητοποίηση καταφέρνει το πρώτο πλήγμα στα σχέδιά μας. Από την άλλη βέβαια είναι κι αυτή η λεπτομέρεια με την κρίση, η οποία εδώ και τρία περίπου χρόνια μας κοπανάει αλύπητα επειδή ως λαός είμαστε τεμπέληδες και κομπιναδόροι, οπότε καλά να πάθουμε τώρα. Δεύτερο χτύπημα αυτό και σε λίγο το κακό τριτώνει, όταν μετά από απλές αριθμητικές πράξεις αποδεικνύεται ότι το ταξιδάκι μας θέλει τον προϋπολογισμό πολλών, μα πάρα πολλών μελλοντικών τευχών του deltaHacker.



Και να λοιπόν, αγαπητές κι αγαπητοί, άλλος ένας από τους άπειρους τρόπους που έχει η πραγματικότητα για να συγκρούεται με την επιθυμία, να ανατρέπει τα πάντα και γενικώς να κάνει τα σχέδιά μας μαντάρα. Για να είμαι όμως δίκαιος και να μη λαϊκίζω, οφείλω να σημειώσω ότι στον συγκεκριμένο υποχώρο της πραγματικότητας υπάρχει σε περίσσεια, ευτυχώς, το στοιχείο «μου αρέσει».

Δεν εννοοώ φυσικά ότι μου αρέσει η κρίση, ούτε το ότι το δρομολόγιο που φτιάξαμε με τη @citr0nella πρέπει να πάρει μια κάποια, εχμ, μικρή αναβολή. Αυτό που εννοώ είναι ότι, με δεδομένες τις εξαιρετικά δύσκολες και δυσμενείς συνθήκες, κάποιοι από εμάς πήραμε την απόφαση να κάνουμε αυτό που μας αρέσει κι όχι αυτό που μας είπαν ότι πρέπει να κάνουμε[1]. Είναι βλέπετε πεποίθησή μας ότι για να πετύχει μια προσπάθεια, απαραίτητη προϋπόθεση είναι να σ’ αρέσει αυτό που κάνεις. Αποτελεί επίσης θεάρεστο έργο το να προσπαθείς να δημιουργείς τις προϋποθέσεις για να κάνεις αυτό που σου αρέσει — εκτός κι αν σου αρέσει κάτι άτιμο ή/και ανήθικο, δηλαδή.

Σ’ εμάς, που λέτε, αρέσει να ασχολούμαστε με θέματα σαν αυτά που διαβάζετε κάθε μήνα στο περιοδικό. Ακριβώς γι’ αυτό, πριν δύο περίπου χρόνια αποφασίσαμε να πάμε κόντρα σ’ όλα τα «πρέπει» και να δημιουργήσουμε τις προϋποθέσεις που θα μας επιτρέπουν να κάνουμε ό,τι αγαπάμε. Κάπως έτσι γεννήθηκε το deltaHacker κι από τότε είμαστε όλοι πιο χαρούμενοι, που κάθε μήνα φροντίζουμε και σας το ταχυδρομούμε. Και ξέρετε γιατί; Μα, γιατί όταν κάνεις κάτι με όρεξη και χαρά, εκτός του ότι περνάς καλά, το πώς αισθάνεσαι φαίνεται παραέξω, εκτιμάται και με τον έναν ή τον άλλον τρόπο ανταμείβεται. Επιπρόσθετα, επιτρέπει στη δουλειά σου να ξεχωρίζει από παρόμοιες δουλειές άλλων, οι οποίοι κάνουν ό,τι κάνουν επειδή είναι αναγκασμένοι κι όχι επειδή τους αρέσει.

Τελικά, δεν ξέρω αν πιάσατε το μήνυμα, αλλά ούτε φέτος έχει ταξίδι στο Ρέικιαβικ. Χρόνια το σχεδιάζω, μ’ αρέσει ο Βορράς, μ’ αρέσουν τα πολικά φώτα, μ’ αρέσει και το γεγονός ότι μιλάμε για την πρωτεύουσα ενός μικρού νησιού, στη μέση του ωκεανού. Αλλά, όλα κι όλα, πιο πολύ μ’ αρέσει ν’ ασχολούμαι με το περιοδικό. Βλέπω μάλιστα ότι αρέσει και σ’ εσάς το περιοδικό. Και το ότι σας αρέσει, με κάνει να μ’ αρέσει πιο πολύ αυτό που κάνω. Φαύλος κύκλος — αλλά από εκείνους που δεν θέλεις να τους ξεφύγεις! Η Σύμβουλος Εκδόσεως (ΤΜ) συμφωνεί μαζί μου — και ξέρω ότι τις ίδιες απόψεις συμμερίζονται κι όλοι οι φίλοι και συνεργάτες του περιοδικού.

Αλλά ας σας αφήσω να απολαύσετε τη δουλειά τους για το τεύχος.

1. Βλέπε editorial του deltaHacker 001.

σνέιλ μέιλ



Αγαπητοί μου,

Λοιπόν, αρχίζω από τα τυπικά. Χρόνια πολλά και καλά, καλές γιορτές, καλή Σαρακοστή, άντε και με το καλό να έλθει κι από τα μέρη μας η Ανάστασις, να έλθει και το άγιον Φως να μας φωτίσει.

Κυριακή 31 Μαρτίου, σήμερα, και προ ολίγου μιλάγαμε με τον ανιψιό στο σκάιπ, δηλαδή η Μαιρούλα του μίλαγε στο σκάιπ και από δίπλα ήμασταν όλοι, εγώ, η εξαδέλφη, η μητέρα της εξαδέλφης, και η κυρία Καίτη.

Έχουν Πάσχα εκεί στα ξένα, μας έλεγε, φέτος έπεσε πολύ νωρίς, το ακριβώς αντίθετο από μας, που φέτος πέφτει Μάιο μήνα. Έχουν αργία, λοιπόν. Τετραήμερο.

«Και γιατί δεν ερχόσουνα μια που έχετε τετραήμερο;» τον ρώτησε σε κάποια στιγμή η Κυρία Καίτη.

Φαίνεται όμως ότι μια από τις παράλληλες απώλειες της οικονομικής κρίσης είναι το κόστος των εισιτηρίων. Πού τα παλιά τα χρόνια που με 100 ευρώπουλα πήγαινες και ερχόσουνα Αγγλία μια χαρά. Σήμερα το φτηνότερο που βρίσκεις αρχίζει από τα 250 και φτάνει ένας Θεός ξέρει πού. Στο τέλος, θα καταλήξουμε να θέλεις να πας και συ σαν άνθρωπος στο Χάρροντ'ς για τις εκπτώσεις και δεν θα συμφέρει.

Η ζωή ακριβαίνει στο Ηνωμένο Βασίλειο, αγαπητοί μου. Πάρτε, ας πούμε, το κόστος των σπουδών. Το πανεπιστήμιο, λέει, κάνει γύρω στις 9 χιλιοπούλες τον χρόνο. Λίρες. Και στο ποσό δεν περιλαμβάνεται καν το κόστος στέγης, διατροφής και μετακινήσεων.

Τα μεταπτυχιακά είναι «φτηνότερα», γύρω στις 4 με πέντε χιλιάδες, λίρες πάντα. Plus φως νερό τηλέφωνο.

Μιλάμε για χοντρά ποσά.

Περιέργως πώς, ωστόσο, το ύψος διδάκτρων και κόστους ζωής δεν φαίνεται να επηρεάζει καθόλου τα λιοκαμμένα βλαστάρια της πατρίδας μας. Κρίση, ξε-κρίση, αυτά καταπλέουν μαζικά στα Βρετανικά Πανεπιστήμια. Κι όταν λέω μαζικά, το εννοώ. Μαζικά. Σαν τα λέμμινγκς ένα πράγμα.

Σύμφωνα με τις επίσημες στατιστικές, το 2012 ήμασταν οι τέταρτοι από τους Ευρωπαίους σε /απόλυτους/ αριθμούς. Πρώτοι οι Γερμανοί, μετά Ιρλανδοί και Γάλλοι, μετά εμείς, κι από πίσω κολλητά οι Κύπριοι. Οι υπόλοιποι, ξέρετε, Ισπανοί, Ιταλοί και λοιποί, να ακολουθούν στο βάθος.

Η έφεση του Έλληνα για μάθηση, βλέπετε.

Ναι, καλά. Μάθηση.

Κοιτάξτε, μπορεί να κάνω και λάθος, αλλά αυτό που έχω καταλάβει είναι το εξής: Οι Έλληνες φοιτητές στο εξωτερικό πρωτίστως περνάνε καλά. Οι σπουδές έπονται.

Βασικά κάνουν παρέα μόνο με Έλληνες, συγκατοικούν και βγαίνουν μόνο με Έλληνες, ακούνε ελληνικούς σταθμούς στο ίντερνετ, ή ρεμπέτικα και Αλ-

κίνοο Ιωαννίδη, τρώνε σπιτικό φαγητό που κουβαλάνε με τάπερ από την Ελλάδα, βρίζουν τον καιρό, λένε πόσο φοβερή και τρομερή είναι η Ελλάδα και οι Έλληνες, και πόσο μίζεροι και ξενέρωτοι είναι όλοι οι άλλοι, και γενικώς φτιάχνουν ένα ελληνοπρεπές theme park και κλείνονται μέσα.

Βγαίνουν μόνο όταν τους επισκέφτεται, στα μέρη τους, κανένας Έλληνας καλλιτέχνης ή πολιτικός, για να μιλήσει στην τάδε ή την δείνα εκδήλωση, ή όταν παίζει καμιά ελληνική ταινία σε κανένα ψαγμένο σινεμά. Τότε τους πιάνει το σύνδρομο του απόδημου ελληνισμού και πάνε όλοι για μπούγιο.

Αυτό που προσπαθώ να πω είναι πως το ένα πράγμα που /δεν/ έχω δει να κάνουν οι Έλληνες φοιτητές στο εξωτερικό είναι να ανοίγονται για να μάθουν κάτι από την χώρα που τους φιλοξενεί.

Δεν μιλάω για το αντικείμενο των σπουδών τους. Αυτό κουτσά στραβά θα το μάθουν. Αλλά αυτό, και μέσω σκάιπ πάλι θα το μάθαιναν. Δεν χρειαζόταν να πάνε σε άλλη χώρα.

Μιλάω για τον τρόπο ζωής. Για την ιστορία. Για την οργάνωση του κράτους. Για τον σεβασμό στον πολίτη. Για ό,τι καλό τέλος πάντων έχει να επιδείξει μια χώρα, η κάθε χώρα. Κάτι τις θα έχει, δεν μπορεί.

Θυμάμαι όταν έφευγε ο ανιψιός για Λονδίνο, προ δεκαετίας σχεδόν, τον είχε πάρει κατά μέρος η μητέρα της εξαδέλφης, που είναι Ινδουίστρια, και του είχε δώσει μια συμβουλή. «Ό,τι κι αν κάνεις εκεί στα ξένα», του είχε πει, «προσπάθησε ένα πράγμα: Μην κάνεις το λάθος να μπερδέψεις την νοσταλγία με την αδιαφορία.»

«Μα τι λέτε», είχε τσινήσει αυτός. «Ποια νοσταλγία; Ποια αδιαφορία; Ακόμα δεν έφυγα.»

«Ξέρω τι σου λέω», επέμεινε αυτή. «Τόσοι και τόσοι πάνε έξω και μουλαρώνουν. Το ένα τους μυρίζει, το άλλο τους βρωμάει. Τους πιάνει νοσταλγία κι αυτοί νομίζουν ότι τους ενοχλεί η χώρα. Τυφλώνονται. Νομίζουν ότι τα ξέρουν όλα. Ότι τα καταλαβαίνουν όλα. Όχι. Τίποτα δεν ξέρουν. Όταν πας σε μια ξένη χώρα πρέπει να είσαι ανοιχτός, έτοιμος να τα μάθεις όλα από την αρχή. Αυτό μας διδάσκει ο Κρίσνα.»

Ο ανιψιός είχε τσινήσει, αλλά εκ των υστέρων παραδέχεται ότι η συμβουλή, αν και υπερβολική, ήταν εν πολλοίς σωστή.

Διότι μόνο αν πάψεις να πιστεύεις ότι τα ξέρεις όλα, μπορείς να ανοίγεις τα μάτια σου για να δεις πόσα πράγματα δεν ήξερες.



Λοιπόν, με αυτό συμφωνώ απολύτως. Είναι γενικότερο. Δεν έχει να κάνει μόνο με την ζωή στο εξωτερικό.

Να σας φέρω ένα λίγο διαφορετικό παράδειγμα. Λέμε εδώ στην Ελλάδα και ξαναλέμε για την φιλοσοφία, και τον Σωκράτη, και τον Πλάτωνα, και τον Αριστοτέλη, και όλους αυτούς. Πόσο σπουδαίοι και τρανοί είναι και λοιπά. Και είναι πράγματι σπουδαίοι και τρανοί. Αλλά δεν θα μπορέσουμε ποτέ να συλλάβουμε πόσο σπουδαίοι και τρανοί είναι αυτοί οι αρχαίοι γίγαντες, αν δεν μάθουμε τι έχουν κάνει οι επόμενοι γίγαντες, όσοι εμπνεύστηκαν από το έργο των αρχαίων.

Καταλαβαίνεις καλύτερα την παγκόσμια σημασία του Αριστοτέλη και του Πλάτωνα όταν, ας πούμε, διαβάζεις για το έργο του Αβικέννα ή του Αβερρόη.

Όταν, ας πούμε, διαβάζεις ότι ο Αβικέννας ήταν αυτός που ξαναέφερε τους ευρωπαίους μελετητές του μεσαίωνα σε επαφή με την αρχαία ελληνική σκέψη, ή όταν μαθαίνεις πως με αφορμή τον σχολιασμό του Αριστοτέλη που έκανε ο Αβερρόης ξεκίνησε ο «διάλογος» των αρχαίων Ελλήνων με τον Θωμά τον Ακινάτη ή τον Ιωάννη Ντανς Σκώτο, και άνοιξε ο δρόμος για τον Καρτέσιο και τη σύγχρονη δυτική φιλοσοφία.

Αβικέννας ή Ιμπν Σίνα. Ο σημαντικότερος εκπρόσωπος της ισλαμικής φιλοσοφίας. Από την Περσία. Πλήρες όνομα: Αμπού Αλή αλ-Χουσεΐν ιμπν Αμπντ Άλλα ιμπν Σίνα.

Αβερρόης ή Ιμπν Ρουσντ. Ο πρώτος συστηματικός σχολιαστής του έργου του Αριστοτέλη, θεωρούμενος από πολλούς ως πατέρας της κοσμικής (δηλαδή μη θεολογικής) φιλοσοφίας στην Μεσαιωνική Ευρώπη. Άραβας από την Κόρντοβα της Ισπανίας. Πλήρες όνομα: Αμπού Ι-Ουαλίντ Μουχάμαντ ιμπν Αχμάντ ιμπν Ρουσντ.

Θα μας διαβάζει κανένας από τα Χρυσά Αβγά και θα τον έχει πιάσει αλλεργικό σοκ.

Μάλιστα, αγαπητοί μου.

Γίγαντας ο Αριστοτέλης, αλλά γίγαντας και ο Αβικέννας. Γίγαντας ο Πλάτωνας αλλά γίγαντας και ο Αβερρόης και ο Καρτέσιος και ο Καντ και τόσοι άλλοι.

Το να πιστεύουμε, ή να συμπεριφερόμαστε σαν να πιστεύουμε ότι οι αρχαίοι Έλληνες είχανε πει τα πάντα, και ότι κανείς άλλος μετά από εκείνους δεν αξίζει της προσοχής μας, είναι κοντόφθαλμο, επικίνδυνο και κατά βάση παθολογικό.

Και λέω παθολογικό γιατί ένα βηματάκι μόνο μας χωρίζει από όσους πιστεύουν ότι η Μέρκελ και οι Εβραιομασώνοι μάς ψεκάζουν για να μας καθυποτάξουν, και νομίζουν ότι αυτά που ζούμε στην Ελλάδα (ή στην Κύπρο) οφείλονται σε ξένα κέντρα εξουσίας που μας επιβουλεύονται γιατί είμαστε οι καλύτεροι και γιατί οι θάλασσές μας είναι γεμάτες υδατάνθρακες.

Θα μου πείτε, οι ψεκασμοί κι οι υδατάνθρακες μας μάραναν;

Εδώ φτάνουμε να ακούμε τον Πρωθυπουργό της χώρας να μας διαβεβαιώνει με δελτίο τύπου ότι το DNA των Ελλήνων, το γονίδιό μας, έχει πολύ ισχυρά αντισώματα που πολεμούν τον "ιό" (sic) του ρατσισμού, του αντισημιτισμού και του νεοναζισμού, και δεν αντιλαμβανόμαστε την αντίφαση, την ευήθεια, για να μην πω κάτι χειρότερο.

Οι ψεκασμοί είναι το λιγότερο.

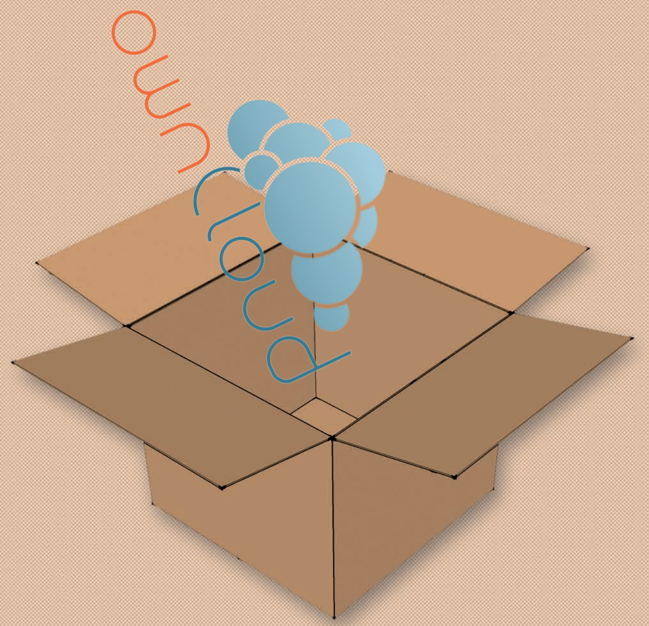
Τέλος πάντων. Αυτά. Καλή Σαρακοστή.

Σας ασπάζομαι

Θείος Ακάκιος

Skill: Beginner

Tags: Ubuntu, ownCloud, cloud storage, file synchronization



Poorman's Dropbox

by GiannouG

Το Dropbox του φτωχού!

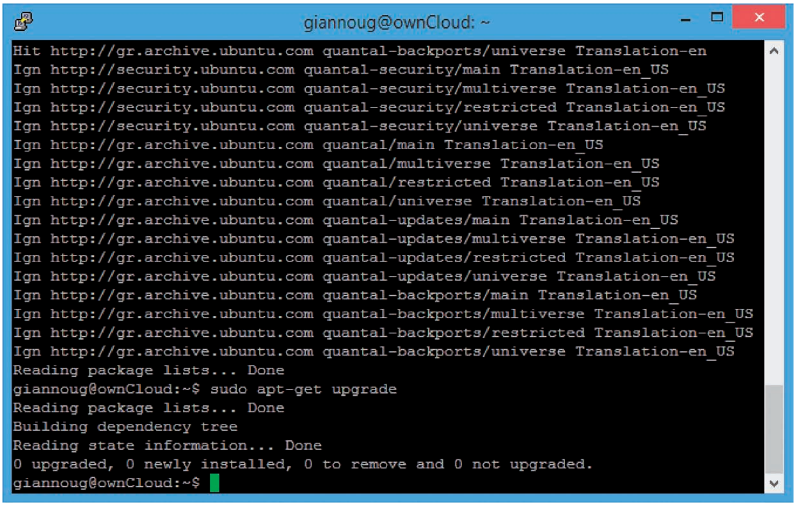
Το cloud αποτελεί το μέλλον στην αποθήκευση δεδομένων. Αυτό πιστεύουν όσοι έχουν δοκιμάσει το Dropbox ή και άλλες υπηρεσίες, που προσφέρουν αποθηκευτικό χώρο *κάπου* στο Internet. Μια δοκιμή θα πείσει κι εσάς γι' αυτό! Γιατί όμως να εμπιστευτούμε τα αρχεία μας σε *τρίτους* και γιατί να πληρώσουμε για λίγο παραπάνω χώρο, τη στιγμή που μπορούμε να στήσουμε μόνοι μας ανάλογες υπηρεσίες;

Δεν είναι λίγες οι φορές που ακούμε για επιθέσεις σε υπηρεσίες που προσφέρουν αποθηκευτικό χώρο, ενώ μερικές φορές μαθαίνουμε ότι διέρρευσαν passwords ή ακόμη και αρχεία χρηστών! Πέρα εξάλλου από τις επιθέσεις, μια μικρή απερισκεψία από έναν υπάλληλο είναι αρκετή για να εκτεθούν χιλιάδες χρήστες στο έλεος κάποιου περίεργου, που ήταν στο σωστό μέρος, τη σωστή στιγμή. Όταν όμως έχουμε εμείς οι ίδιοι τον έλεγχο του αποθηκευτικού χώρου, αν μη τι άλλο νιώθουμε περισσότερο ασφαλείς. Εξάλλου, έχοντας τον πλήρη έλεγχο του σχετικού συστήματος, μπορούμε να προσθέτουμε αποθηκευτικό χώρο αλλά και νέες δυνατότητες κατά βούληση. Φυσικά, το δικό μας σύστημα δεν θα είναι σώνει και καλά πιο ασφαλές από εκείνα των εμπορικών υπηρεσιών, από την άλλη όμως είναι λιγότερο πιθανό να αποτελέσει στόχο για τους επιτήδειους.

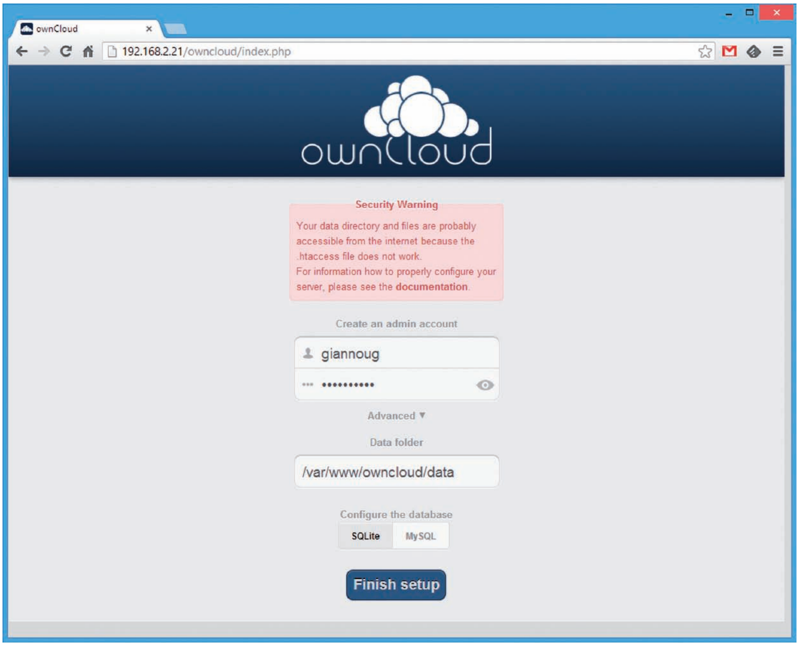
Επιτρέψτε μας να σας συστήσουμε το ownCloud! Πρόκειται για μια εφαρμογή Open Source που προσφέρει τις ίδιες δυνατότητες —και σε ορισμένες περιπτώσεις παραπάνω— με εκείνες γνωστών υπηρεσιών δικτυακής αποθήκευσης. Το ownCloud μάς παρέχει τη δυνατότητα εύκολης δημιουργίας ενός δικτυακού αποθηκευτικού χώρου, ο οποίος μάλιστα θα υποστηρίζει και το λεγόμενο «συγχρονισμό» μεταξύ των υπολογιστών μας. Επιπλέον, το ownCloud είναι δυνατόν να πλαισιωθεί από διάφορα πρόσθετα, τα οποία προσδίδουν δυνατότητες που δεν θα γνωρίσει ποτέ του το Dropbox και οι άλλες εμπορικές υπηρεσίες. Προσοχή: Το ownCloud προσφέρει *και* εμπορικές υπηρεσίες, μέσα από το `http://owncloud.com`. Εμείς αναφερόμαστε στο λογισμικό που διατίθεται δωρεάν, στο `http://owncloud.org`.

Προετοιμασία

Για το ownCloud θα χρειαστούμε *φυσικά* έναν Linux server! Το αν θα είναι κάποιος server στο υπόγειο, κάποιο VPS στη μακρινή Κουάλα Λουμπούρ ή ένα VM στο ίδιο μας το PC, δεν παίζει κανένα ρόλο. Εμείς, πάντως, για τις ανάγκες του άρθρου σηκώσαμε το ownCloud σε ένα ταπεινό VM. Τώρα, θέλουμε για λίγο την προσοχή σας: Επειδή ο γράφων δεν θέλει να παίζει με την ασφάλεια των δεδομένων του, προτίμησε να κρύψει το σύστημα με το ownCloud στην ασφάλεια του τοπικού του δικτύου, πίσω



Το σύστημά μας θέλουμε να είναι αναβαθμισμένο στην τελευταία έκδοση του Ubuntu 12.10. Στην συνέχεια θα εγκατα-στήσουμε το repository του ownCloud.



Η εγκατάσταση έλαβε τέλος και το ownCloud είναι πλέον έτοιμο! Θα χρειαστεί να δώσουμε το όνομα χρήστη του διαχειριστή, να δηλώσουμε το που επιθυμούμε να αποθηκεύονται τα δεδομένα αλλά και τον τύπο της βάσης δεδομένων.

από το firewall του DSL router. Θα σας προτείναμε να κάνετε κι εσείς το ίδιο. Αυτή η διευθέτηση περιορίζει κάπως την ευκολία, αλλά δεν την καταργεί! Από το τοπικό δίκτυο θα μπορούμε να χρησιμοποιούμε άνετα τις υπηρεσίες του προσωπικού μας cloud, ενώ όταν βρισκόμαστε εκτός θα έχουμε άνετη πρόσβαση μέσω OpenVPN[1]. Στην εικονική μηχανή με το ownCloud εγκαταστήσαμε το Ubuntu 12.10 αν και, τουλάχιστον θεωρητικά, θα μπορούσαμε να είχαμε στραφεί σε οποιαδήποτε άλλη διανομή. Αρχικά, ξεκινήσαμε εξασφαλίζοντας ότι το σύστημα είναι πλήρως αναβαθμισμένο.

```
sudo apt-get update
sudo apt-get upgrade
```

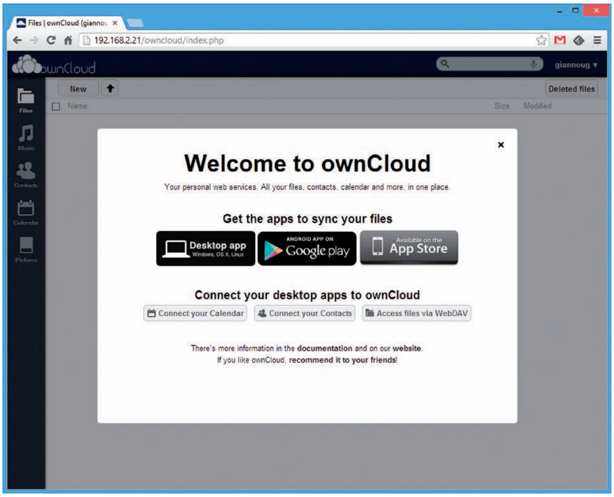
Για την εγκατάσταση του ownCloud προτιμήσαμε τα πακέτα που προσφέρουν οι προγραμματιστές του και όχι εκείνα της διανομής μας. Βλέπετε, τα repositories της διανομής διαθέτουν παλιότερη έκδοση του ownCloud κι αυτό δεν μας άρεσε καθόλου. Έτσι, συνδεθήκαμε στο σύστημα, αποκτήσαμε δικαιώματα root και προσθέσαμε τα repositories (αποθετήρια) του ownCloud. Αμέσως μετά το εγκαταστήσαμε το ownCloud χρησιμοποιώντας το apt-get.

```
sudo su
echo 'deb \
http://download.opensuse.org/repositories/
isv:ownCloud:community/xUbuntu_12.10/ /' >> \ /etc/apt/sources.
list.d/owncloud.list
wget \
http://download.opensuse.org/repositories/
```

1. Βλέπε σχετικό άρθρο στο deltaHacker 007 ή το αντίστοιχο άρθρο στο http://deltahacker.gr/?p=5669.

```
isv:ownCloud:community/xUbuntu_12.10/Release.key
apt-key add - < Release.key
apt-get update
apt-get install owncloud
```

Η εγκατάσταση ολοκληρώθηκε χωρίς καμία επιπλέον ενέργεια από μέρους μας.



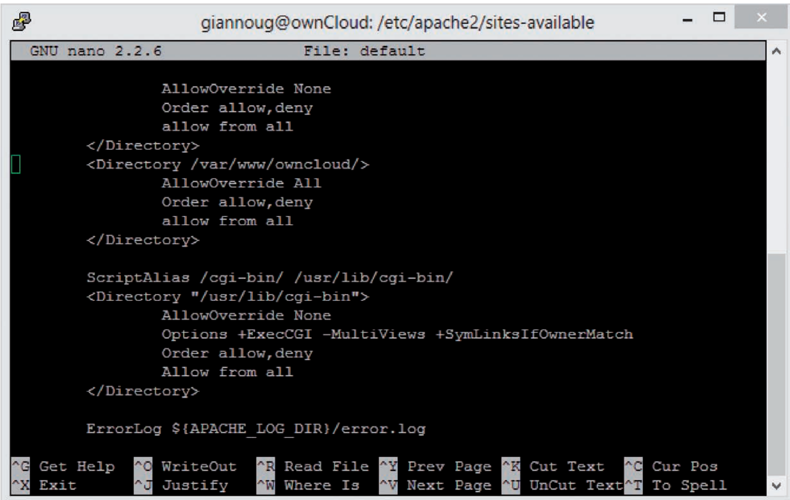
Welcome to ownCloud! Για αρχή, το σύστημα μας προτείνει να κατεβάσουμε τις ειδικές εφαρμογές συγχρονισμού των αρχείων μας και να τις εγκαταστήσουμε στους υπολογιστές μας.

Πάνω στο σύννεφο

Μόλις τελειώσει η εγκατάσταση με το apt, το ownCloud θα είναι διαθέσιμο στο http://ip/owncloud. Προφανώς, στη θέση του «ip» μπαίνει η διεύθυνση IP του μηχανήματος. Αρχικά, το σύστημα θα μας κάνει διάφορες ερωτήσεις: Ποιο είναι το όνομα χρήστη του διαχειριστή για το ownCloud; Πού θέλουμε να αποθηκεύονται τα δεδομένα; Ποια βάση δεδομένων θέλουμε να χρησιμοποιηθεί; Εμείς δίνουμε τα επιθυμητά στοιχεία για τον διαχειριστή του ownCloud (π.χ., το προσωπικό μας username κι ένα καλό password), αφήνουμε την προεπιλεγμένη διαδρομή για την αποθήκευση των αρχείων κι επιλέγουμε για βάση το SQLite. Για την ώρα, αγνοούμε τυχόν μηνύματα που μας προειδοποιούν ότι τα αρχεία είναι προσβάσιμα από το δίκτυο. Πρόκειται για bug το οποίο θα διορθώσουμε αργότερα και (ελπίζουμε) να διορθωθεί κι επίσημα στην επόμενη έκδοση του ownCloud.

Αφού ξεμπερδέψουμε με τα παραπάνω, θα μας υποδεχθεί ένα μεγάλο παράθυρο καλωσορίσματος. Μέσα από αυτό, το σύστημα θα μας προτείνει να εγκαταστήσουμε τα προγράμματα συγχρονισμού, τόσο στο PC, όσο και στο smartphone. Επιπρόσθετα, παρέχεται η επιλογή συγχρονισμού των επαφών και του ημερολογίου μας.

Το μενού του ownCloud διαθέτει πέντε κουμπιά και το καθένα αντιστοιχεί σε έναν διαφορετικό τύπο αρχείου. Εξαίρεση αποτελεί το πρώτο κουμπί, το οποίο εμφανίζει *όλα* τα αρχεία που έχουμε ανεβάσει στο σύννεφό μας. Ακολουθούν κουμπιά για τη μουσική, τις επαφές, τις εγγραφές του ημερολογίου και τις φωτογραφίες. Περιττό να πούμε ότι πατώντας κάποιο κουμπί το ownCloud δεν περιορίζεται στην απλή εμφάνιση των αντίστοιχων αρχείων, αλλά προσφέρει και ορισμένες σχετικές λειτουργίες. Έτσι, στην περίπτωση της μουσικής μπορούμε να ακούσουμε τα τραγούδια μας απευθείας μέσα από τον Browser, ενώ στην περίπτωση των φωτογραφιών, μπορούμε να τις παρακολουθήσουμε σε slide show! Κάπου εδώ μπορούμε να κάνουμε ένα μικρό διάλειμμα από την εργασία μας και να περιηγηθούμε στο περιβάλλον του ownCloud. Πάντως η αλήθεια είναι ότι δεν έχουμε και πολλές επιλογές. Το User Interface είναι αρκετά απλό! Στα δεξιά της οθόνης θα βρούμε το μενού του χρήστη, καθώς και μερικές επιλογές του συστήματος. Στο dropdown μενού που εμφανίζεται μπορούμε να αλλάξουμε τις προσωπικές μας ρυθμίσεις, να προσθέσουμε νέους



```
giannoug@ownCloud: /etc/apache2/sites-available
GNU nano 2.2.6                 File: default

                AllowOverride None
                Order allow,deny
                allow from all
        </Directory>
        <Directory /var/www/owncloud/>
                AllowOverride All
                Order allow,deny
                allow from all
        </Directory>

        ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
        <Directory "/usr/lib/cgi-bin">
                AllowOverride None
                Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
                Order allow,deny
                Allow from all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log

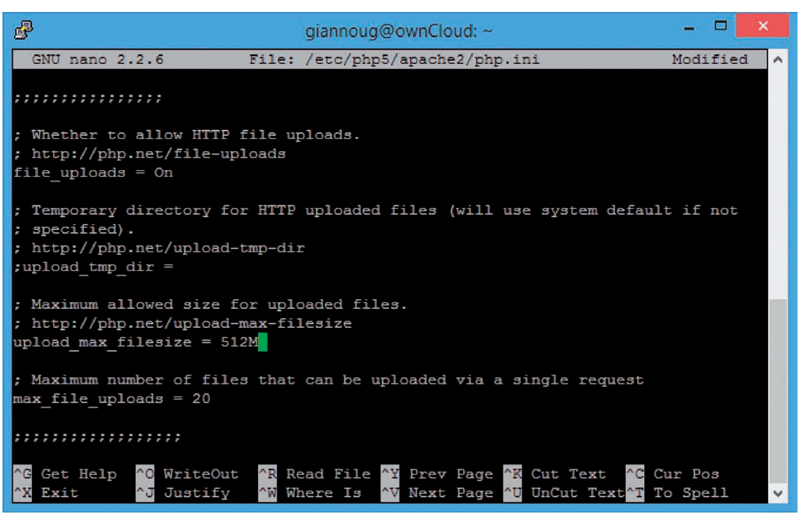
^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Αρχικά θα διορθώσουμε ένα bug που συνοδεύει την τρέχουσα έκδοση του ownCloud. Δε θέλουμε να βλέπει κανένας μη εξουσιοδοτημένος χρήστης τα αρχεία μας!

χρήστες, να διαχειριστούμε τις εφαρμογές που έχουμε εγκαταστήσει στο ownCloud και, φυσικά, να αποσυνδεθούμε από το λογαριασμό μας. Για τις εφαρμογές θα μιλήσουμε σε λίγο. Για την ώρα, καλό θα ήταν να παίξουμε με τις υπόλοιπες επιλογές.

Οι πρώτες τροποποιήσεις

Μη μας παρεξηγείτε! Πιστεύουμε ότι το ownCloud είναι αρκετά καλό και *σχεδόν* άξιος αντικαταστάτης των... ξέρετε ποιών! Υπάρχουν όμως δύο μικρές αλλαγές που πιστεύουμε ότι μπορούν να βελτιώσουν τη λειτουργία του συστήματος. Για αρχή, πρέπει να διορθώσουμε το bug που αναφέραμε προηγουμένως και το οποίο επιτρέπει σε *όλους* να βλέπουν τα αρχεία μας. Το συγκεκριμένο bug αναμένεται να διορθωθεί πολύ σύντομα και δεν πρέπει να παραξενευτείτε αν απουσιάζει από τη δική σας εγκατάσταση. Σε κάθε περίπτωση, το εν λόγω πρόβλημα αντιμετωπίζεται



```
giannoug@ownCloud: ~
GNU nano 2.2.6          File: /etc/php5/apache2/php.ini              Modified

;;;;;;;;;;;;;;;;

; Whether to allow HTTP file uploads.
; http://php.net/file-uploads
file_uploads = On

; Temporary directory for HTTP uploaded files (will use system default if not
; specified).
; http://php.net/upload-tmp-dir
;upload_tmp_dir =

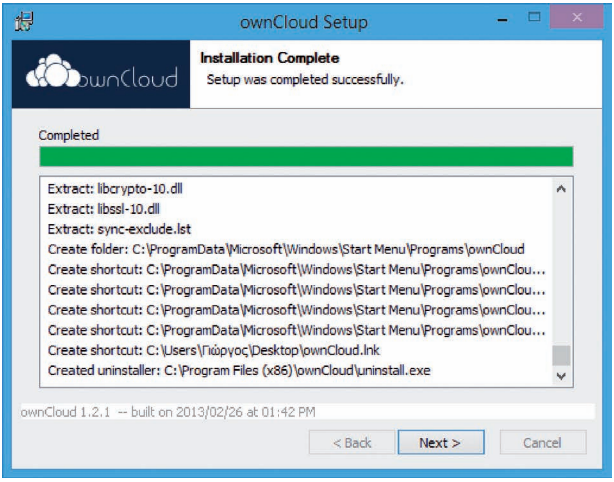
; Maximum allowed size for uploaded files.
; http://php.net/upload-max-filesize
upload_max_filesize = 512M

; Maximum number of files that can be uploaded via a single request
max_file_uploads = 20

;;;;;;;;;;;;;;;;;;

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

Ποιος ανέχεται τον περιορισμό των 2MB για το ανέβασμα αρχείων; Η σχετική επέμβαση γίνεται από το αρχείο php.ini κι εμείς ανεβάσαμε το όριο στα 512MB.



Η εγκατάσταση της εφαρμογής του ownCloud είναι εξαιρετικά απλή: Next και πάλι Next!

πολύ εύκολα. Αρκεί να δηλώσουμε στον Apache ότι *επιτρέπεται* να αλλάζουν οι ρυθμίσεις του μέσα από τα αρχεία .htaccess. Αρχικά, λοιπόν, θα χρειαστεί να ανοίξουμε το αρχείο /etc/apache2/sites-available/default. Εκεί, μέσα στο <VirtualHost *:80> προσθέτουμε το ακόλουθο μπλοκ ρυθμίσεων:

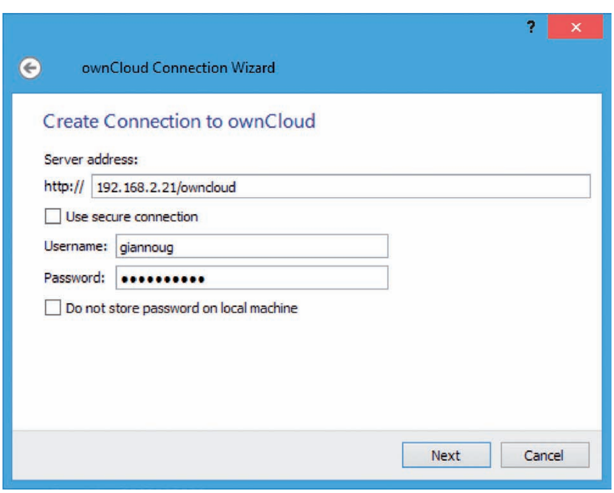
```
<Directory /var/www/owncloud/>
  AllowOverride All
  Order allow,deny
  allow from all
</Directory>
```

Για να λάβει υπ' όψιν του τις αλλαγές ο Apache, θα χρειαστεί να τον επανεκκινήσουμε. Εκτελούμε την παρακάτω εντολή και αμέσως μετά επισκεπτόμαστε το Administration panel του ownCloud, για να δούμε αν εξακολουθεί να εμφανίζεται το σφάλμα. (Όχι, ε; :D)

```
sudo service apache2 restart
```

Κάτι ακόμα που μας ενοχλεί στις «εργοστασιακές» ρυθμίσεις, είναι το μέγιστο επιτρεπόμενο μέγεθος για τα αρχεία που ανεβάζουμε. Ποιος ο λόγος να περιοριζόμαστε στα 2MB; Για να αλλάξουμε αυτό το όριο, πρέπει να πειράξουμε τις ρυθμίσεις της PHP! Πρόκειται για το php.ini, που βρίσκεται στον κατάλογο /etc/php5/apache2. Η γραμμή που πρέπει να αλλάξουμε στο εσωτερικό του είναι εκείνη με τη ρύθμιση upload_max_filesize[2]. Εμείς επιλέξαμε μια τιμή στα 512MB, αφού δεν πρόκειται να ανεβάσουμε ποτέ κάτι μεγαλύτερο. Μετά από αυτή την επέμβαση, απαιτείται μια ακόμα επανεκκίνηση στον Apache. Πλέον, αν επισκεφθούμε την αρχική σελίδα του ownCloud και αφήσουμε το δείκτη του ποντικιού πάνω στο κουμπί του upload, θα εμφανιστεί ένα κουτάκι που θα μας ενημερώνει ότι το μέγιστο μέγεθος των αρχείων είναι 512MB!

Για να συνδεθούμε στο ownCloud μέσω του προγράμματος συγχρονισμού, πρέπει να δώσουμε τη διεύθυνση της υπηρεσίας (το σχετικό URL) όπως επίσης το όνομα χρήστη και το συνθηματικό μας.



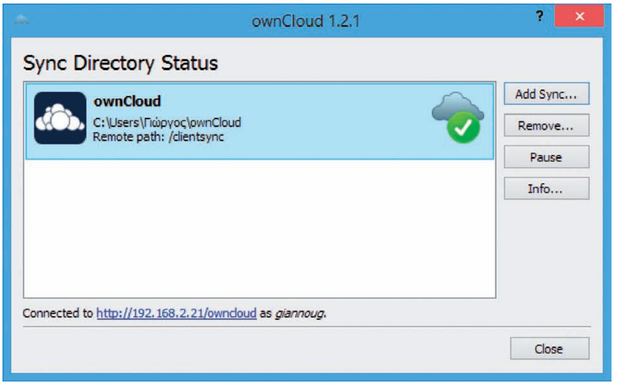
Συγχρονισμός αρχείων a la Dropbox

Μια υπηρεσία φιλοξενίας αρχείων δεν είναι τίποτα χωρίς το πρόγραμμα που φροντίζει να ανεβάζει τα αρχεία μας στο σύννεφο και, φυσικά, να τα κατεβάζει, όταν τα τοπικά αντίγραφα είναι παλιότερα από εκείνα στο χώρο της υπηρεσίας. Οι developers του ownCloud έχουν δημιουργήσει εφαρμογές για κάθε πλατφόρμα, που πετυχαίνουν όλα τα προαναφερθέντα. Με λίγα λόγια, φροντίζουν για τον λεγόμενο «συγχρονισμό» των αρχείων του υπολογιστή μας με εκείνα στο

2. Αν ανοίξουμε το αρχείο με το nano, μπορούμε να αναζητήσουμε τη γραμμή πατώντας [Ctr +W].

ownCloud. Για το άρθρο θα δούμε τη λειτουργία της εφαρμογής στα Windows. Τουλάχιστον θεωρητικά, δεν θα υπάρχει καμία διαφορά με τις εφαρμογές για το Linux και το OS X.

Το πρόγραμμα συγχρονισμού μπορούμε να το κατεβάσουμε από το https://owncloud.org. Όπως πάντα στα Windows, η εγκατάσταση γίνεται πατώντας μερικές φορές το Next. Οι προκαθορισμένες ρυθμίσεις είναι μια χαρά και δεν χρειάζεται να αλλάξουμε καμία. Στο τέλος θα μας ζητηθεί να δώσουμε ορισμένες πληροφορίες για την εγκατάσταση του ownCloud μας. Πιο συγκεκριμένα, θα χρειαστεί να δώσουμε τη διεύθυνσή του (το URL που χρησιμοποιούμε για σύνδεση από τον browser), το όνομα χρήστη και τον κωδικό μας. Πλέον, στο «home directory» του υπολογιστή μας θα υπάρχει ένας φάκελος με το όνομα ownCloud. Οτιδήποτε προσθέσουμε εκεί θα ανεβαίνει αυτόματα στο ownCloud! Σας θυμίζει κάτι αυτή η λειτουργία;

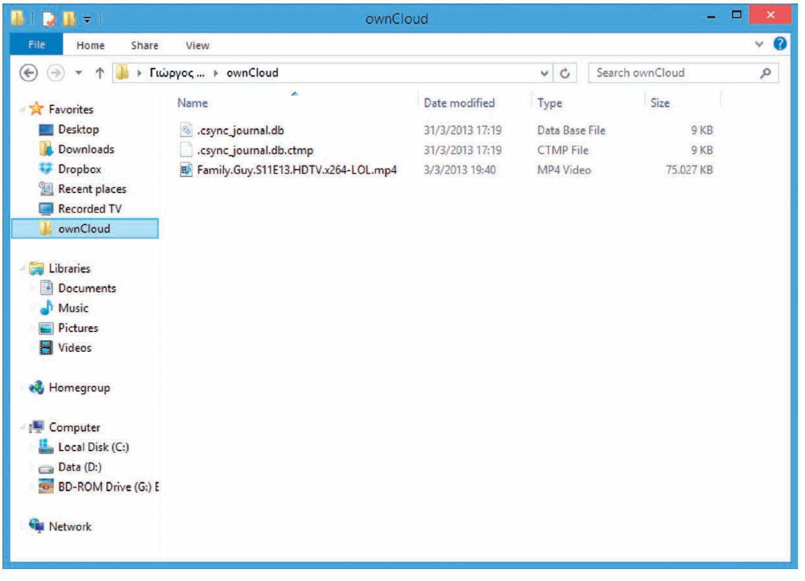


Μπορούμε να εγκαταστήσουμε την εφαρμογή του ownCloud σε όσους υπολογιστές θέλουμε! Επιπρόσθετα, μπορούμε να βλέπουμε ποιοι υπολογιστές έχουν συνδεθεί με το λογαριασμό μας και, αν το επιθυμούμε, να διακόπτουμε ή να παύουμε το συγχρονισμό τους.

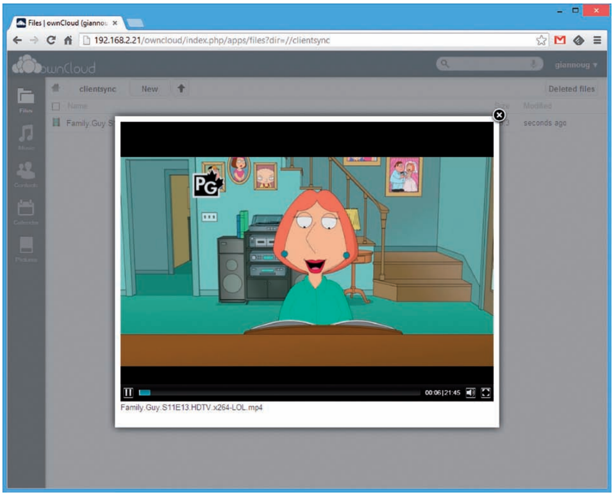
Εξίσου απλά μπορούμε να εγκαταστήσουμε την εφαρμογή σε όσους ακόμη υπολογιστές θέλουμε, ανεξαρτήτως λειτουργικού και αρχιτεκτονικής. Δε θυμόμαστε αν το αναφέραμε πριν, αλλά το ownCloud υποστηρίζει *και* smartphones!

Πρόσθετες λειτουργίες

Ένα από τα μεγαλύτερα προσόντα του ownCloud είναι η εύκολη επέκταση, με την προσθήκη ειδικών plugins. Για παράδειγμα, μπορεί κάποιος να ενσωματώσει στο ownCloud τον δικό του mail client. Κάποιος άλλος θα μπορούσε να εγκαταστήσει ένα πρόσθετο που επιτρέπει στο



Για να ανεβάσουμε ένα αρχείο, αρκεί να το αντιγράψουμε στον φάκελο που δημιούργησε το πρόγραμμα συγχρονισμού.



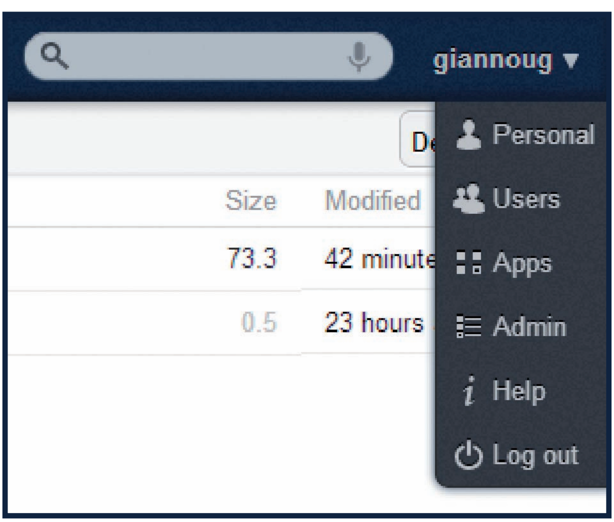
*Αφού ανεβεί το αρχείο μας, αυτόματα θα κατεβεί σε όλα τα PC όπου έχουμε το πρόγραμμα συγχρονισμού ήδη εγκατεστημένο. Επιπλέον, μπορούμε μέσω της σελίδας να δούμε το αρχείο μας. Για τις δοκιμές μας ανεβάσαμε ένα βίντεο και, όπως βλέπετε, μπορούμε να το δούμε *και* online!*

ownCloud να κατεβάζει *αυτόματα* τα torrent files που ρίχνουμε σε προκαθορισμένο φάκελο του υπολογιστή μας. Πώς σας φαίνονται αυτά; Για να δούμε μια λίστα με όλα τα διαθέσιμα plugins (η επίσημη ονομασία τους είναι «apps»), δεν έχουμε παρά να επισκεφθούμε το web interface του onwCloud. Από εκεί πάμε στο dropdown μενού στα δεξιά κι επιλέγουμε το Apps. Έτσι, στην αριστερή περιοχή του interface θα εμφανιστούν όλες οι εφαρμογές που είναι ενεργοποιημένες ή εγκεκριμένες από την ομάδα του ownCloud. Για να ενεργοποιήσουμε κάποια, αρκεί να κάνουμε πάνω της ένα κλικ! Μια αρκετά ενδιαφέρουσα εφαρμογή είναι η External storage support, που επιτρέπει την προσθήκη εξωτερικών μέσων αποθήκευσης στο ownCloud. Εδώ όμως υπάρχει μια ενδιαφέρουσα λεπτομέρεια: Με τον όρο «εξωτερικός χώρος» εννοούνται *και* οι FTP servers, αλλά και κάποιος λογαριασμός στο Dropbox ή σε παρόμοια υπηρεσία. Αρκετά έξυπνο για όσους τα θέλουν όλα και μάλιστα... σε ένα!

Επιπλέον εφαρμογές μπορούμε να βρούμε πατώντας στο κουμπί More Apps, στο επάνω μέρος της λίστας. Αυτή που μας τράβηξε αρκετά το ενδιαφέρον είναι η RuTorrent. Νομίζουμε ότι υποψιάζεστε τι κάνει και δεν χρειάζεται να μπούμε σε λεπτομέρειες.

Επίλογος

Τελικά, με μια εικονική μηχανή ή ένα παλιό μηχάνημα, μπορούμε να φτιάξουμε τη δική μας εκδοχή του Dropbox και μάλιστα χωρίς πολύ κόπο. Το ownCloud επιτρέπει να πετύχουμε αυτό που οι σχετικές υπηρεσίες στο Internet θα χρέωναν αρκετά, ενώ μας δίνει και το περιθώριο να προσαρμόσουμε τη συμπεριφορά του στις απαιτήσεις μας. Ποιος θα φανταζόταν ότι μπορούμε να κατεβάσουμε torrents μέσω του ownCloud; Το Dropbox πάντως σίγουρα δε μπορεί :P



Στο μενού στα δεξιά εμφανίζονται οι επιλογές για το σύστημα, το προφίλ μας αλλά και τις πρόσθετες εφαρμογές που μπορούμε να εγκαταστήσουμε στο ownCloud.

openSUSE Conference

18-22 July 2013

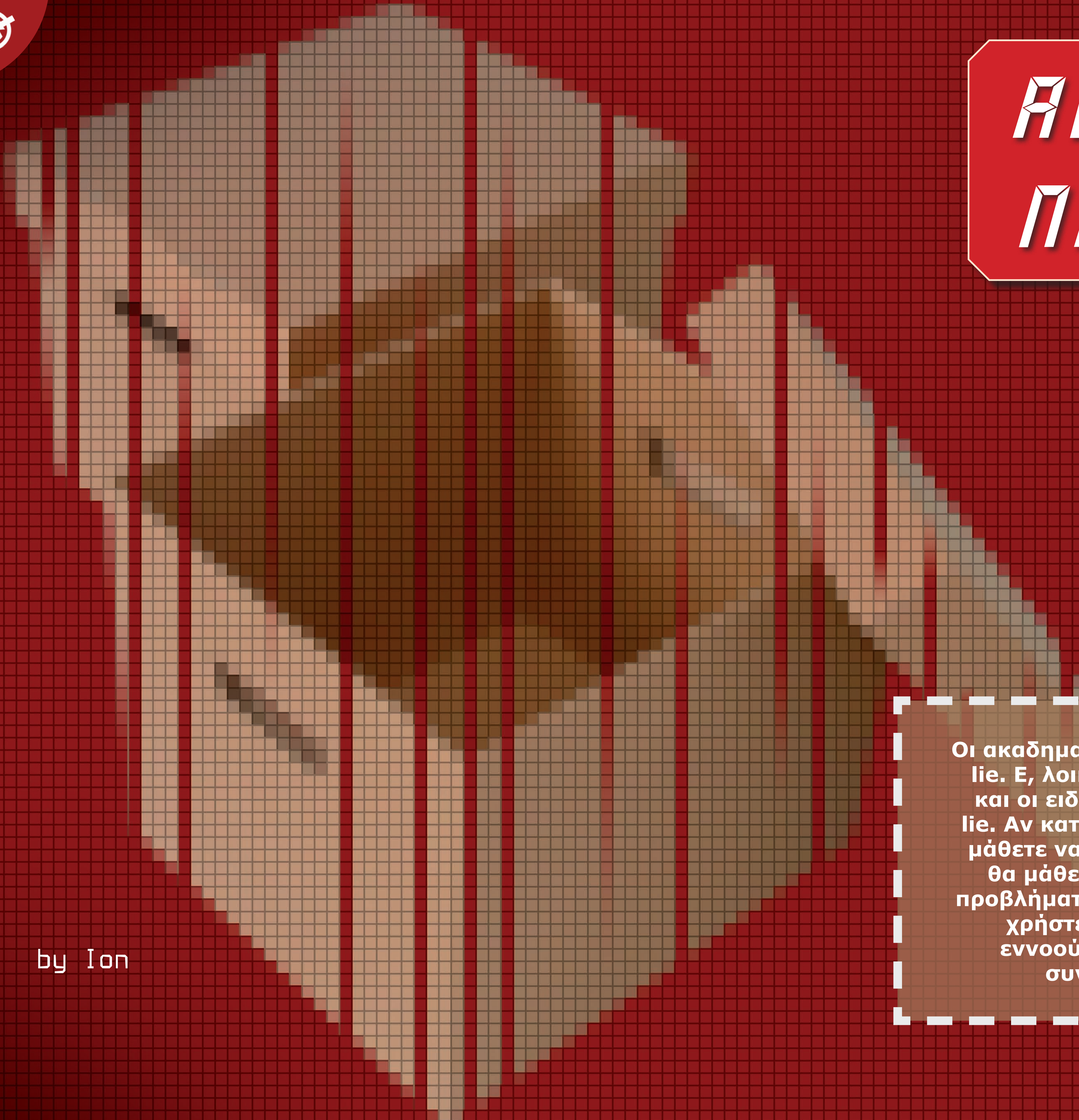
Thessaloniki, Greece

Olympic Museum

See you there !!!

conference.opensuse.org

#opensuse-conference @ Freenode



ΑΝΑΛΥΣΗ ΠΑΚΕΤΩΝ

[μέρος 1]

Skill: Beginner

Tags: tcpdump, packets, sniffing, capturing, extraction, analysis

Οι ακαδημαϊκοί έχουν ένα ρητό: Data don't lie. Ε, λοιπόν, ένα παρόμοιο ρητό έχουν και οι ειδικοί ασφαλείας: Packets don't lie. Αν κατανοήσετε αυτή την αλήθεια και μάθετε να την αξιοποιείτε, αν μη τι άλλο θα μάθετε να αντιμετωπίζετε ύπουλα προβλήματα ή να εντοπίζετε κακόβουλους χρήστες. Θέλετε να καταλάβετε τι εννοούμε; Καθίστε αναπαυτικά και συνεχίστε την ανάγνωση...

by Ion

Τι πηγαίνει στραβά με το δίκτυο; Αν ξέρατε πόσες φορές έχει προκύψει αυτή η ερώτηση στα μυαλά χιλιάδων διαχειριστών συστημάτων ανά τον κόσμο… Η αλήθεια είναι ότι σ' ένα δίκτυο υπάρχουν πολλά πράγματα που μπορούν να πάνε στραβά και πολλές φορές είναι αδύνατο να αντιληφθούμε τι ακριβώς προκαλεί το πρόβλημα. Αντί να φιλοσοφούμε ή να πιθανολογούμε για το τι πηγαίνει στραβά, είναι πολύ προτιμότερο να αναζητήσουμε τα αίτια στην πηγή. Μ' αυτό δεν εννοούμε τον router ή κάποια άλλα συσκευή αλλά τα δικτυακά πακέτα (network packets), που αποτελούν τους φορείς επικοινωνίας μεταξύ των δικτυακών συσκευών. Σε αυτό το άρθρο (όπως και στα επόμενα) θα ασχοληθούμε με την ανάλυση πακέτων ή αλλιώς packet analysis. Μια στιγμή, όμως! Πριν προσπαθήσετε να αναλύσετε τη λειτουργία ενός δικτύου, είναι απαραίτητο να γνωρίζετε πώς ακριβώς λειτουργεί. Γι' αυτό το λόγο, αν σας λείπουν βασικές γνώσεις για τη λειτουργία των δικτύων, καλό θα ήταν να μελετήσετε πριν προχωρήσετε παρακάτω. Μια καλή ιδέα θα ήταν να διαβάσετε τα άρθρα της σειράς *Πώς δουλεύουν τα δίκτυα*[1]. Όπως βλέπετε έχουμε φροντίσει για όλα — γι' αυτό άλλωστε μας αγαπάτε ;)

Ανάλυση Πακέτων

Η ανάλυση πακέτων (packet analysis) αποτελεί τη διαδικασία κατά την οποία «ακούμε» και καταγράφουμε τη ροή των πακέτων που διέρχονται από κάποιο network interface του υπολογιστή μας. Σκοπός μας είναι η ανάλυση τις καταγεγραμμένης ροής πακέτων, για να αντιληφθούμε τι συμβαίνει στο δίκτυο. Η ανάλυση πακέτων μπορεί να βοηθήσει στα ακόλουθα ζητήματα:

- Κατανόηση των χαρακτηριστικών ενός δικτύου
- Καταγραφή όλων των συσκευών που βρίσκονται σε χρήση εντός του δικτύου
- Προσδιορισμός των συσκευών ή των υπηρεσιών που προκαλούν προβλήματα ή ευθύνονται για το φόρτο στο δίκτυο
- Καταγραφή των χρονικών περιόδων κατά τις οποίες έχουμε σημαντική αύξηση της κίνησης
- Αναγνώριση πιθανών επιθέσεων ή κακόβουλης δραστηριότητας
- Εντοπισμός εφαρμογών που έχουν κενά ασφαλείας ή προκαλούν συμφόρηση

Η καταγραφή και η ανάλυση πραγματοποιείται από ειδικά εργαλεία που ονομάζονται packet sniffers. (Ο γράφων έχει δει τον αντίστοιχο ελληνικό όρο να αναγράφεται ως «ωτακουστές πακέτων» — και φυσικά ανακατεύτηκε). Η λειτουργία αυτών των προγραμμάτων μπορεί να χωριστεί σε τρεις διαφορετικές φάσεις.

1. **Συλλογή των δεδομένων.** Στο πρώτο βήμα ο packet sniffer καταγράφει δεδομένα (πακέτα) από το ενσύρματο ή ασύρματο δίκτυο. Αυτό συνήθως επιτυγχάνεται ενεργοποιώντας το promiscuous mode στην εκάστοτε κάρτα δικτύου. Αν δεν έχετε ξανακούσει τον όρο, σημαίνει πως η κάρτα δικτύου θα ακούει (και άρα θα μπορεί να καταγράφει) όλη τη δικτυακή κίνηση και όχι μόνο αυτήν που προορίζεται για τη συγκεκριμένη κάρτα.

1. Τεύχη 004, 005, άρθρο στο http://deltahacker.gr/?p=5138 και τεύχος 018.

2. **Μετατροπή σε κατανοητή μορφή.** Στο δεύτερο βήμα, τα δεδομένα που συγκεντρώθηκαν μετατρέπονται σε μορφή «human readable», όπως λέμε. Δηλαδή, σε μια κατανοητή από τον άνθρωπο μορφή. Φυσικά, ακόμα κι έτσι δεν είμαστε σε θέση να βγάλουμε πολλά συμπεράσματα.
3. **Ανάλυση.** Στο τρίτο και τελευταίο βήμα πραγματοποιείται η ανάλυση των παραπάνω δεδομένων. Ο packet sniffer μπορεί να πάρει τα δεδομένα που καταγράφηκαν, να τα κατηγοριοποιήσει με βάση το πρωτόκολλό τους και να εξάγει κάποια πρώτα στοιχεία. Το υπόλοιπο (και χρονοβόρο) κομμάτι της ανάλυσης, καθώς και τα τελικά συμπεράσματα, θα εξαχθούν από τον αναλυτή.

Τα σύνεργα

Όπως αναφέρθηκε, για την ανάλυση πακέτων προσφέρονται αρκετά εξειδικευμένα εργαλεία. Δύο από τα πιο γνωστά, με τα οποία θα ασχοληθούμε κιόλας, είναι το tcpdump και το Wireshark. Εδώ αξίζει να αναφέρουμε και το TShark, το οποίο αποτελεί μια παραλλαγή του Wireshark χωρίς GUI. Στο παρόν άρθρο θα ασχοληθούμε με το tcpdump. Πρόκειται για ένα μάλλον ιστορικό εργαλείο, του οποίου η πρώτη έκδοση γράφτηκε το 1987! Από τότε αναπτύσσεται κανονικά και η τελευταία έκδοση κυκλοφόρησε τον Ιούνιο του 2012. Η επίσημη ιστοσελίδα βρίσκεται στη διεύθυνση www.tcpdump.org. Μαζί με το tcpdump διατίθεται και η βιβλιοθήκη libpcap, στην οποία βασίζεται και το Wireshark. Στη διαδικασία που θα γνωρίσουμε, θα αξιοποιήσουμε και διάφορα μικρά προγράμματα τα οποία είναι ενσωματωμένα στις περισσότερες διανομές Linux. Εμείς πάντως εργαστήκαμε με τη διανομή BackTrack και συγκεκριμένα με την έκδοση 5.0 R3. Σε αυτήν το tcpdump είναι ήδη εγκατεστημένο. Εσείς, ωστόσο, μπορείτε να χρησιμοποιήσετε οποιαδήποτε διανομή σας βολεύει (π.χ., το τελευταίο Ubuntu). Σε κάθε περίπτωση, η εγκατάσταση του tcpdump είναι παιχνιδάκι. Σε διανομές Debian-based, όπως είναι το Ubuntu, αρκεί να δώσουμε κάτι τέτοιο:

```
root@bt:~# apt-get install tcpdump
```



Το βασικό εργαλείο της δουλειάς: Το tcpdump και η βιβλιοθήκη στην οποία βασίζεται (libpcap). Πρόκειται για το καλύτερο όπλο στην ανάλυση πακέτων...

Tcpdump (gotta catch 'em all!)

Σημειώστε ότι για να τρέξει επιτυχώς το tcpdump πρέπει να το εκτελέσουμε είτε από τον λογαριασμό του root είτε με τη βοήθεια της εντολής sudo. Ας δούμε για αρχή ποια είναι τα διαθέσιμα network interfaces που μπορούμε να χρησιμοποιήσουμε με το tcpdump:

```
root@bt:~# tcpdump -D
1.eth0
2.usbmon1 (USB bus number 1)
3.any (Pseudo-device that captures on all interfaces)
4.lo
```

Όπως βλέπουμε, στον υπολογιστή από τον οποίο δουλεύουμε είναι τέσσερα. Τα πιο χρήσιμα είναι το πρώτο και το τρίτο. Το interface eth0 αντιστοιχεί συνήθως στην ενσύρματη κάρτα δικτύου του υπολογιστή μας, ενώ αυτό το any είναι μια ψευδο-συσκευή που αναπαριστά όλα τα διαθέσιμα network interfaces. Έτσι, αν έχουμε δύο ενσύρματες και μια ασύρματη κάρτα δικτύου, μπορούμε να καταγράψουμε όλα τα πακέτα που διέρχονται από όλες τις συσκευές χρησιμοποιώντας την ψευδο-συσκευή any. Για να δηλώσουμε στο tcpdump το επιθυμητό interface χρησιμοποιούμε την παράμετρο -i. Ας κάνουμε μία πρώτη δοκιμή:

```
root@bt:~# tcpdump -i eth0 -n

tcpdump: verbose output suppressed, use -v or -vv for full
protocol decode

listening on eth0, link-type EN10MB (Ethernet), capture size
65535 bytes
```

Αν περιμένετε λίγο, θα αρχίσετε να βλέπετε διάφορα μηνύματα. Μπορείτε να σταματήσετε τη διαδικασία πατώντας τον συνδυασμό πλήκτρων [CTRL+C]. Κάθε μία από τις γραμμές που εμφανίζονται αποτελεί στην ουσία τις πληροφορίες για τη μετακίνηση ενός πακέτου, μέσω του υπό εξέταση interface. Το συντακτικό του tcpdump, με το οποίο εμφανίζονται όλες οι εγγραφές, ακολουθεί το εξής μοτίβο:

```
[timestamp] [πρωτόκολλο] [IP αφετηρίας].[port αφετηρίας] > [IP
προορισμού].[port προορισμού][περιεχόμενο πακέτου]
```

Για παράδειγμα, ας δούμε τις ακόλουθες εγγραφές από τη δική μας καταγραφή, αφού δοκιμάσαμε να επισκεφτούμε την ιστοσελίδα του περιοδικού:

```
21:46:05.382099 IP 192.168.1.118.59658 > 192.168.1.1.53: 52825+
A? deltahacker.gr. (32)

21:46:05.383530 IP 192.168.1.1.53 > 192.168.1.118.59658: 52825
1/0/0 A 109.74.192.50(48)
```

Η πρώτη γραμμή λέει ότι στις 21:46:05 (και κάτι milliseconds), χρησιμοποιώντας το πρωτόκολλο IP ο υπολογιστής με διεύθυνση 192.168.1.118 (ο δικός μας) έστειλε ένα πακέτο από τη θύρα 59658 στη θύρα 53 του υπολογιστή με IP 192.168.1.1 (o router). Παραδίπλα βλέπουμε τι ήταν αυτό που του έστειλε, όπου φαίνεται να ρώτησε: «ποιο είναι το στοιχείο A για το deltahacker.gr». Τι καταλαβαίνουμε απ' αυτό; Εδώ είναι που χρειάζονται οι γνώσεις περί δικτύων. Όπως είναι γνωστό, η θύρα 53 χρησιμοποιείται από την υπηρεσία DNS. Το Domain Name System δεν κάνει τίποτε άλλο από το να μετατρέπει τα φιλικά προς τον χρήστη domain names (όπως το deltahacker.gr) στις αντίστοιχες διευθύνσεις IP. Ένας DNS server περιέχει διάφορες εγγραφές που ονομάζονται records και είναι διαφόρων τύπων (NS, MX κ.ά.). Ένας από αυτούς τους τύπους είναι και ο A, που στην ουσία αντιστοιχεί στη διεύθυνση IP του εκάστοτε domain name. Εν ολίγοις, για να μπορέσει ο υπολογιστής μας να επικοινωνήσει με το site του deltaHacker πρέπει να μάθει ποια είναι η IP διεύθυνση του server που το φιλοξενεί. Έτσι, λοιπόν, στέλνει ένα αίτημα προς τη συσκευή με IP 192.168.1.1 (πρόκειται για τον router, που εκτελεί και χρέη DNS server), ζητώντας το A record του deltahacker.gr (ζητώντας, δηλαδή, το IP του). Στη δεύτερη γραμμή βλέπουμε την απάντηση του router, ο οποίος στέλνει στον υπολογιστή μας ένα πακέτο με τη διεύθυνση 109.74.192.50. Από εδώ και στο εξής, ο υπολογιστής μας είναι σε θέση να επικοινωνήσει με το website του περιοδικού.

Ας δούμε κι ένα δεύτερο παράδειγμα. Αν αφήσουμε το tcpdump να καταγράφει, σύντομα θα δούμε εγγραφές σαν τις ακόλουθες:

```
21:46:06.229431 ARP, Request who-has 192.168.1.118 tell
192.168.1.1, length 46

21:46:06.229447 ARP, Reply 192.168.1.118 is-at
08:00:27:28:00:59, length 28
```

Χμ, εδώ τα πράγματα είναι κάπως πιο απλά. Στην πρώτη γραμμή έχουμε ένα request του πρωτοκόλλου ARP που λέει: «πείτε σας παρακαλώ στον υπολογιστή/συσκευή με IP 192.168.1.1 ποιος υπολογιστής του δικτύου έχει την IP 192.168.1.118». Ακριβώς από κάτω έχουμε μια απάντηση σε αυτό το αίτημα, που χρησιμοποιεί πάλι το πρωτόκολλο ARP και λέει το εξής: «ο υπολογιστής με IP 192.168.1.118 βρίσκεται στο 08:00:27:28:00:59». Αναρωτιέστε τι συμβαίνει εδώ; Αυτά που μόλις είδαμε αποτελούν δύο πακέτα του πρωτοκόλλου ARP (Address Resolution Protocol). Στην ουσία πρόκειται για ένα πρωτόκολλο που χρησιμοποιείται για την επικοινωνία δύο υπολογιστών, οι οποίοι βρίσκονται στο ίδιο δίκτυο. Βλέπετε, δύο υπολογιστές του ίδιου δικτύου δεν ανταλλάσσουν πακέτα χρησιμοποιώντας τις διευθύνσεις IP. Αντίθετα, χρησιμοποιούν τις διευθύνσεις MAC των δικτυακών καρτών. Το πρωτόκολλο ARP μετατρέπει τις διευθύνσεις IP των υπολογιστών του δικτύου στις αντίστοιχες διευθύνσεις MAC. Αυτό το κάνει στέλνοντας ένα πακέτο σε όλο το δίκτυο ρωτώντας: «Ποιος υπολογιστής έχει την τάδε διεύθυνση IP;» Ο υπολογιστής με αυτήν την IP, χρησιμοποιώντας και πάλι το ARP, απαντάει και δηλώνει την αντίστοιχη διεύθυνση MAC. Αν θυμόσαστε, στο ARP έχουμε αναφερθεί ήδη σε διάφορα άρθρα (βλ. π.χ. εκείνα στο τεύχος 001) αλλά και στο 5ο επεισόδιο της 1ης σεζόν του deltaCast (http://deltahacker.gr/?p=7869), όπου εξηγούμε κι επιδεικνύουμε πώς δουλεύει η περιβόητη επίθεση του ARP poisoning!

Όπως βλέπετε, για να ερμηνεύετε τα αποτελέσματα μιας δικτυακής καταγραφής χρειάζονται ορισμένες γνώσεις και λίγη εμπειρία γύρω από το πώς λειτουργούν τα δίκτυα και οι γνωστές δικτυακές υπηρεσίες. Όμως, πώς αλλιώς θα τις αποκτήσετε αν δεν βουτήξετε στα βαθιά νερά; Νομίζουμε ότι η διαδικασία του packet sniffing που εξετάζουμε θα βοηθήσει αρκετά.

Εξαγωγή πληροφοριών από packet capture file, παράδειγμα 1

Ας προχωρήσουμε τώρα στις υπόλοιπες δυνατότητες του tcpdump. Μία από τις πιο σημαντικές είναι αυτή της αποθήκευσης των αποτελεσμάτων της καταγραφής, σε αρχεία τύπου PCAP. Αυτό μπορούμε να το επιτύχουμε με την παράμετρο -w.

```
root@bt:~# tcpdump -i eth0 -n -w out.pcap

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture
size 65535 bytes

[…]

^C
47618 packets captured

47618 packets received by filter

0 packets dropped by kernel
```

```
root@bt:~# tcpdump -n -r out.pcap 'tcp or udp' | cut -d " " -f 3 | cut -d "." -f 1-4 | sort -nr | uniq
reading from file out.pcap, link-type EN10MB (Ethernet)
212.205.159.144
212.205.159.141
212.155.198.30
194.63.247.30
194.63.247.155
194.63.247.138
192.168.1.118
192.168.1.1
109.74.192.50
95.211.169.84
93.186.137.146
68.232.35.121
31.13.84.8
fe80::807d:645e:d92c:25.546
root@bt:~#
```

Το αποτέλεσμα της εξαγωγής διευθύνσεων IP από το αρχείο μας, αφού φιλτράραμε τα αποτελέσματα για να είναι ευανάγνωστα!

Όπως βλέπετε, σαν αρχείο εξόδου ορίσαμε το out.pcap και μετά από λίγη ώρα, αφού αφήσαμε το tcpdump να καταγράψει την κίνηση για ένα εύλογο χρονικό διάστημα, σταματήσαμε τη διαδικασία πατώντας [CTRL+C]. Αφού αποθηκεύσουμε τη δικτυακή κίνηση σε ένα αρχείο, μπορούμε να τη μελετήσουμε με το ίδιο tcpdump. Παράλληλα, με τη χρήση διαφόρων βοηθητικών προγραμμάτων για τη γραμμή εντολών του Linux, μπορούμε να κρατήσουμε τις πληροφορίες που μας ενδιαφέρουν. Ας φορτώσουμε καταρχήν το αρχείο PCAP στο tcpdump:

```
root@bt:~# tcpdump -n -r out.pcap
```

Μόλις εκτελέσετε το παραπάνω, θα δείτε αμέσως έναν χαμό από πακέτα — όπως ακριβώς θα τα βλέπατε κατά τη ζωντανή καταγραφή. Προφανώς, αυτός ο τρόπος προβολής δεν είναι ιδιαίτερα βολικός για να αντλήσουμε δεδομένα. Σε αυτό το σημείο θα μας βοηθήσουν διάφορα άλλα εργαλεία του Linux. Ένα πρώτο βήμα θα ήταν, π.χ., να βγάλουμε από το αρχείο μια λίστα με όλες τις διευθύνσεις IP που συμμετέχουν στην ανταλλαγή πακέτων. Για να πετύχουμε κάτι τέτοιο, αρκεί να χρησιμοποιήσουμε το πρόγραμμα cut:

```
root@bt:~# tcpdump -n -r out.pcap | cut -d " " -f 3
```

Με την παραπάνω εντολή θα πάρετε μια λίστα με όλες τις διευθύνσεις IP, καθώς και τις αντίστοιχες θύρες. Παρατηρείστε ότι στην ουσία ανακατευθύναμε την έξοδο του tcpdump στο πρόγραμμα cut. Οι δύο παράμετροι στο cut λένε το εξής: «Θεώρησε ότι κάθε γραμμή χωρίζεται σε επιμέρους τμήματα με διαχωριστικό σύμβολο (delimiter) το κενό (-d « «) και στη συνέχεια εμφάνισε από κάθε γραμμή το τρίτο πεδίο (-f 3)». Δεν έχουμε τελειώσει όμως! Εμείς θέλαμε μόνο τις διευθύνσεις IP κι όχι τις αντίστοιχες θύρες. Επομένως, πρέπει να ανακατευθύνουμε την προηγούμενη έξοδο (και πάλι) στο πρόγραμμα cut:

```
root@bt:~# tcpdump -n -r out.pcap | cut -d " " -f 3 | cut -d "." -f 1-4
[…]
192.168.1.118
192.168.1.118
109.74.192.50
192.168.1.118
109.74.192.50
```

```
192.168.1.118
109.74.192.50
192.168.1.118
109.74.192.50
192.168.1.118
109.74.192.50
192.168.1.118
109.74.192.50
192.168.1.118
109.74.192.50
192.168.1.118
193.104.215.61
192.168.1.118
193.104.215.61
192.168.1.118
109.74.192.50
109.74.192.50
193.104.215.61
Request
192.168.1.118
109.74.192.50
192.168.1.118
Request
```

Κάθε γραμμή που εισέρχεται εκ νέου στο cut, χωρίζεται σε τμήματα που διαχωρίζονται με την τελεία. Από αυτά, το cut τυπώνει μόνο το πρώτο, το δεύτερο, το τρίτο και το τέταρτο. Όμως ούτε και τώρα έχουμε τελειώσει. Όπως παρατηρείτε, μέσα στις IP διευθύνσεις υπάρχουν και κάποια αλφαριθμητικά (π.χ., το «Request»). Αυτά προέρχονται από γραμμές που περιέχουν μηνύματα του πρωτοκόλλου ARP. Αυτές οι γραμμές είναι περιττές και πρέπει να τις αποκλείσουμε. Αυτό μπορούμε να το επιτύχουμε δίδοντας στο tcpdump μία παράμετρο που καθορίζει το είδος και το επίπεδο του πρωτοκόλλου που θέλουμε να εμφανίζει. Για αυτό το σκοπό μπορούμε να επιστρατεύσουμε τις κατάλληλες λογικές εκφράσεις. Δείτε ένα παράδειγμα και θα καταλάβετε:

```
root@bt:~# tcpdump -n -r out.pcap 'tcp or udp' | cut -d " " -f 3
| cut -d "." -f 1-4
[…]
192.168.1.118
109.74.192.50
192.168.1.118
109.74.192.50
```

```
192.168.1.118
109.74.192.50
192.168.1.118
109.74.192.50
192.168.1.118
193.104.215.61
192.168.1.118
193.104.215.61
192.168.1.118
109.74.192.50
109.74.192.50
193.104.215.61
192.168.1.118
109.74.192.50
192.168.1.118
```

Η λογική έκφραση «tcp or udp» λέει στο tcpdump να διαβάσει από το αρχείο μόνο τις εγγραφές που αφορούν πακέτα των πρωτοκόλλων TCP ή UDP, αγνοώντας τα υπόλοιπα πρωτόκολλα (όπως για παράδειγμα το ARP). Εκτελώντας την παραπάνω εντολή, θα πάρετε μια λίστα με όλες τις διευθύνσεις IP. Σας αρέσει αυτή η λίστα; Σε εμάς πάντως όχι! Βλέπετε, πολλές από τις διευθύνσεις επαναλαμβάνονται αρκετές φορές κι αυτό δεν εξυπηρετεί σε τίποτα — για την ακρίβεια μας μπερδεύει! Για να πετύχουμε κι αυτό το ζητούμενο, θα χρησιμοποιήσουμε δύο ακόμα προγράμματα. Το ένα είναι το sort, το οποίο εμφανίζει ταξινομημένες τις γραμμές που του δίνουμε. Το άλλο είναι το uniq, το οποίο κρατάει κι εμφανίζει μόνο τις μοναδικές γραμμές, απαλείφοντας τα διπλότυπα:

```
root@bt:~# tcpdump -n -r out.pcap 'tcp or udp' | cut -d " " -f 3
| cut -d "." -f 1-4 | sort | uniq
reading from file out.pcap, link-type EN10MB (Ethernet)
109.74.192.50
192.168.1.1
192.168.1.114
192.168.1.118
193.104.215.61
199.66.238.217
62.1.38.10
62.1.38.11
62.1.38.16
68.232.35.121
```

Πολύ καλύτερα! Θέλετε μια άσκηση για το σπίτι; Ψάξτε πώς θα μπορούσατε να αφαιρέσετε από τα παραπάνω αποτελέσματα τις διευθύνσεις που

```
root@bt:~# tcpdump -An -r out.pcap 'dst port 80' | grep Host | sort | uniq
reading from file out.pcap, link-type EN10MB (Ethernet)
Host: 0.gravatar.com
Host: 1.gravatar.com
Host: adman.in.gr
Host: deltahacker.gr
Host: dictionary.in.gr
Host: finance.in.gr
Host: profile.ak.fbcdn.net
Host: secure-uk.imrworldwide.com
Host: static.adman.gr
Host: static.in.gr
Host: www.facebook.com
Host: www.in.gr
root@bt:~#
```

Όλα τα domains που καταγράφτηκαν στο αρχείο PCAP. Αυτός είναι απλός τρόπος να εντοπίσουμε ύποπτες συνδέσεις...

αφορούν στο τοπικό δίκτυο (όσες έχουν την ακόλουθη μορφή: 192.xxx.xxx.xxx).

Εξαγωγή πληροφοριών από packet capture file, παράδειγμα 2

Μία άλλη ενδιαφέρουσα δυνατότητα του tcpdump είναι αυτή που επιτρέπει την εκτύπωση των *περιεχομένων* κάθε πακέτου, σε κώδικα ASCII. Για την ενεργοποίησή της αρκεί να χρησιμοποιήσουμε την παράμετρο –A. Ας δούμε ένα παράδειγμα:

```
root@bt:~# tcpdump -An -r out.pcap
[…]
22:19:04.083188 IP 192.168.1.118.44520 > 62.1.38.10.80: Flags
[.], ack 1818, win 1273, options [nop,nop,TS val 1077518 ecr
270225007], length 0
E..4..@.@......v>.&
...Pgi.*.m)}....&P.....
..q...No
22:19:04.089553 IP 192.168.1.118.44391 > 109.74.192.50.80:
Flags [P.], seq 1476:1828, ack 13655, win 3140, options
[nop,nop,TS val 1077520 ecr 2118429488], length 352
E....[@.@.Hn...vmJ.2.g.P..n?...s...D.!.....
..q.~D.0GET /order/ HTTP/1.1
Host: deltahacker.gr
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:14.0)
Gecko/20100101 Firefox/14.0.1
Accept: text/html,application/xhtml+xml,application/
xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

```
Referer: http://deltahacker.gr/live-stream/
X-Moz: prefetch
22:19:05.359617 IP 192.168.1.118.44391 > 109.74.192.50.80:
Flags [.], ack 15095, win 3320, options [nop,nop,TS val 1077837
ecr 2118432412], length 0
E..4.\@.@.I....vmJ.2.g.P..o...............
..rM~D..
[..]
```

Αν και τα περιεχόμενα πολλών πακέτων δεν παρέχουν άμεσα αξιοποιήσιμες πληροφορίες, υπάρχουν κάποια που παρουσιάζουν ιδιαίτερο ενδιαφέρον. Στο προηγούμενο παράδειγμα, μεταξύ άλλων βλέπουμε τα περιεχόμενα ενός HTTP request προς το site του περιοδικού. Όπως καταλαβαίνετε, εδώ μπορούμε πάλι να φιλτράρουμε τα αποτελέσματα για να κρατήσουμε τα στοιχεία που μας ενδιαφέρουν. Για παράδειγμα, ας υποθέσουμε ότι θέλουμε να δούμε μόνο τα πακέτα του πρωτοκόλλου HTTP (δηλαδή όσα έχουν ως θύρα προορισμού την 80) και, στη συνέχεια, να μάθουμε σε ποια domains αφορούν. Για το πρώτο σκέλος θα δώσουμε στο tcpdump μια (απλή) λογική έκφραση, με την οποία θα καθορίσουμε ποια θύρα προορισμού μας ενδιαφέρει:

```
root@bt:~# tcpdump -An -r out.pcap 'dst port 80'
```

Η έκφραση που χρησιμοποιήσαμε είναι πραγματικά πολύ απλή, αφού δεν περιέχει κανέναν λογικό τελεστή παρά μόνο στο αλφαριθμητικό (string) που μας ενδιαφέρει. Το αποτέλεσμα αυτής της εντολής, όμως, δεν θα είναι και πολύ διαφορετικό από αυτό που είχαμε προηγουμένως: Θα εμφανιστούν χιλιάδες πακέτα, ελάχιστα από τα οποία θα περιέχουν κάτι χρήσιμο. Τουλάχιστον όμως επιτύχαμε τον πρώτο στόχο! Αν παρατηρήσετε τη μορφή ενός HTTP request, θα διαπιστώσετε ότι περιέχει ένα πεδίο με την ονομασία Host, το οποίο δείχνει τη σελίδα στην οποία αποστέλλεται το request. Να λοιπόν το κλειδί για τον δεύτερο στόχο μας. Φιλτράροντας τα προηγούμενα αποτελέσματα για αυτήν τη λέξη, θα κρατήσουμε μόνο τα domain names που ψάχνουμε. Σε αυτή τη δουλειά θα μας βοηθήσει το...

```
root@bt:~# tcpdump -An -r out.pcap 'dst port 80' | grep Host
reading from file out.pcap, link-type EN10MB (Ethernet)
Host: www.facebook.com
Host: ocsp.verisign.com
Host: ocsp.godaddy.com
Host: 79.103.226.150
Host: deltahacker.gr
Host: deltahacker.gr
Host: deltahacker.gr
Host: deltahacker.gr
Host: deltahacker.gr
Host: 1.gravatar.com
Host: 1.gravatar.com
```

```
Host: 0.gravatar.com
Host: 0.gravatar.com
Host: www.in.gr
[…]
```

Ας καθαρίσουμε λίγο τα αποτελέσματα:

```
root@bt:~# tcpdump -An -r out.pcap 'dst port 80' | grep Host |
cut -d ":" -f 2| sort | uniq
reading from file out.pcap, link-type EN10MB (Ethernet)
0.gravatar.com
1.gravatar.com
79.103.226.150
adman.in.gr
bruteforce.gr
deltahacker.gr
dictionary.in.gr
evsecure-ocsp.verisign.com
finance.in.gr
i2.wp.com
ocsp.godaddy.com
ocsp.verisign.com
osarena.net
profile.ak.fbcdn.net
secure-uk.imrworldwide.com
static.adman.gr
static.in.gr
s.ytimg.com
www.facebook.com
www.in.gr
www.osarena.net
www.youtube.com
```

Αυτό ήταν! Αρχικά, καταφέραμε να απομονώσουμε τα ενδιαφέροντα HTTP requests, ενώ στη συνέχεια κρατήσαμε τα αντίστοιχα domain names.

Εξαγωγή αρχείων από packet capture

Ας δούμε τώρα και κάτι πιο ενδιαφέρον. Όπως είδαμε, το tcpdump δεν καταγράφει απλά δικτυακά πακέτα ως γεγονότα, αλλά αποθηκεύει και το περιεχόμενό τους. Αυτή η καταγραφή δεν περιλαμβάνει μόνο τα διάφορα αιτήματα (requests) που περιέχονται σε ένα πακέτο, αλλά και κάθε άλλου είδους δεδομένα. Έτσι, αν έχουμε ένα αρχείο PCAP που περιλαμβάνει

```
root@bt:~# tcpxtract -f out.pcap
Found file of type "png" in session [194.63.247.138:20480 -> 192.168.1.118:64164], exporting to 00000000.png
Found file of type "jpg" in session [212.205.159.144:20480 -> 192.168.1.118:48848], exporting to 00000001.jpg
Found file of type "jpg" in session [212.205.159.144:20480 -> 192.168.1.118:49104], exporting to 00000002.jpg
Found file of type "jpg" in session [212.205.159.144:20480 -> 192.168.1.118:49360], exporting to 00000003.jpg
Found file of type "gif" in session [212.155.198.30:20480 -> 192.168.1.118:7873], exporting to 00000004.gif
Found file of type "jpg" in session [93.186.137.146:20480 -> 192.168.1.118:54743], exporting to 00000005.jpg
Found file of type "jpg" in session [93.186.137.146:20480 -> 192.168.1.118:54487], exporting to 00000006.jpg
Found file of type "jpg" in session [93.186.137.146:20480 -> 192.168.1.118:54231], exporting to 00000007.jpg
root@bt:~#
```

Το εργαλείο tcpxtract συμπληρώνει τις λειτουργίες του tcpdump, αφού μπορεί να εξάγει πολλά από τα αρχεία που μετακινήθηκαν εντός του δικτύου! Στην εικόνα βλέπουμε την εξαγωγή και την αποθήκευση διαφόρων εικόνων…

την καταγεγραμμένη δικτυακή δραστηριότητα μιας χρονικής περιόδου, είμαστε σε θέση να εξάγουμε όλα τα αρχεία που διακινήθηκαν στο δίκτυο! Σε αυτό θα μας βοηθήσει ένα πρόσθετο πρόγραμμα, που ονομάζεται tcpxtract (http://tcpxtract.sourceforge.net). Το tcpxtract λειτουργεί στη γραμμή εντολών, διαβάζει αρχεία PCAP και μπορεί να καταλάβει αν μια συγκεκριμένη ακολουθία πακέτων περιέχει κάποιο αρχείο γνωστού τύπου. Κατόπιν, ακολουθώντας ένα προς ένα τα πακέτα της συγκεκριμένης ακολουθίας, μπορεί να ανασυνθέσει το αρχείο που μεταφέρθηκε. Ας δούμε ένα παράδειγμα, χρησιμοποιώντας και πάλι το out.pcap:

```
root@bt:~# tcpxtract -f out.pcap

Found file of type "png" in session [68.232.35.121:20480 ->
192.168.1.118:60099], exporting to 00000000.png

Found file of type "jpg" in session [212.205.159.144:20480 ->
192.168.1.118:48335], exporting to 00000001.jpg

Found file of type "jpg" in session [212.205.159.144:20480 ->
192.168.1.118:48591], exporting to 00000002.jpg

Found file of type "jpg" in session [212.205.159.144:20480 ->
192.168.1.118:48847], exporting to 00000003.jpg

Found file of type "jpg" in session [212.205.159.144:20480 ->
192.168.1.118:49103], exporting to 00000004.jpg

Found file of type "jpg" in session [212.205.159.144:20480 ->
192.168.1.118:49359], exporting to 00000005.jpg

Found file of type "jpg" in session [212.205.159.144:20480 ->
192.168.1.118:49615], exporting to 00000006.jpg

Found file of type "png" in session [194.63.247.138:20480 ->
192.168.1.118:932], exporting to 00000007.png

Found file of type "gif" in session [212.155.198.30:20480 ->
192.168.1.118:8384], exporting to 00000008.gif

Found file of type "jpg" in session [93.186.137.136:20480 ->
192.168.1.118:9359], exporting to 00000009.jpg

Found file of type "jpg" in session [93.186.137.136:20480 ->
192.168.1.118:9103], exporting to 00000010.jpg

[…]
```

Το πρόγραμμα εξάγει αμέσως όλα τα αρχεία που μπορεί να αναγνωρίσει και τα αποθηκεύει τοπικά. Οι τύποι των αρχείων που αναγνωρίζει το tcpxtract βρίσκονται σε ένα αρχείο ρυθμίσεων, στη διαδρομή /etc/tcpxtract.conf. Σε αυτά περιλαμβάνονται όλοι οι διαδεδομένοι τύποι αρχείων (εικόνες, βίντεο, αρχεία του Office, συμπιεσμένα αρχεία κ.ά.).

Wax on, wax off

Όπως καταλαβαίνετε, ο μόνος τρόπος για να αποκτήσετε τις απαιτούμενες γνώσεις αλλά και μια πολύτιμη εμπειρία γύρω από την ανάλυση πακέτων, είναι η εξάσκηση και η επανάληψη. Κάθε φορά που πραγματοποιείτε μια νέα ανάλυση, θα παρατηρείτε νέα στοιχεία που δεν προσέχατε ως τότε. Είμαστε σίγουροι ότι πολλές φορές θα επιστρέψετε στο τοπικό σας αποθετήριο αρχείων PCAP, για να τσεκάρετε κάτι που υποπτεύεστε ότι σας ξέφυγε. Είπαμε τη λέξη αποθετήριο; Συγγνώμη, αυτό θα αποτελέσει το θέμα ενός επόμενου άρθρου μας :) Ως τότε, όμως, μην καθίσετε με σταυρωμένα χέρια! Κατ’ αρχάς, κατεβάστε το ακόλουθο αρχείο packet capture: `http://bit.ly/dh019pcap`. Δεν πρόκειται για κάτι ιδιαίτερο, αλλά νομίζουμε ότι θα σας βοηθήσει αρκετά να εξασκηθείτε. Αναρωτιέστε τι πρέπει να κάνετε με αυτό; Προτείνουμε τα ακόλουθα:

1. Πόσα πακέτα (ανεξαρτήτως πρωτοκόλλου) περιέχονται στο αρχείο;
2. Πόσα πακέτα του πρωτοκόλλου TCP και UDP περιέχονται στο αρχείο;
3. Ποιες είναι οι καταγεγραμμένες διευθύνσεις IP;
4. Ταξινομείστε τις διευθύνσεις IP, σύμφωνα με το πλήθος των εμφανίσεων τους.
5. Υπάρχουν πακέτα του πρωτοκόλλου POP3; Αν ναι, τι περιέχουν τα emails που ανταλλάχθηκαν;
6. Υπάρχουν πακέτα του πρωτοκόλλου FTP; Αν ναι, ποια usernames και passwords χρησιμοποιήθηκαν;
7. Ποιοι είναι οι hosts των HTTP requests;
8. Πόσα και τι είδους αρχεία μπορούν να εξαχθούν από το PCAP;
9. Εντοπίζετε κάποιο άλλο ενδιαφέρον στοιχείο;

Θα περιμένουμε τις απαντήσεις σας στη fan page μας στο Facebook (`https://facebook.com/deltaHacker`) ή στο κατάλληλο forum, στο site του περιοδικού (`http://deltahacker.gr/forums`) — εννοείται μαζί με τις αντίστοιχες εντολές που χρησιμοποιήσατε :)

Επιλογος

Σε αυτό το άρθρο πήραμε μια πρώτη γεύση από τη διαδικασία του packet sniffing/capturing και της ανάλυσης που ακολουθεί. Πρόκειται για μια πολύ ενδιαφέρουσα και ίσως λίγο επίπονη διαδικασία, η οποία όμως θα μας ανταμείψει με πολλές γνώσεις για τις εσωτερικές λειτουργίες των δικτύων και των πρωτοκόλλων. Όπως καταλαβαίνετε, η υπόθεση έχει αρκετό ψωμί ακόμα! Σε επόμενα άρθρα θα ασχοληθούμε με πιο προχωρημένες δυνατότητες, θα χρησιμοποιήσουμε περισσότερα εργαλεία και θα δούμε πιο χρήσιμα παραδείγματα. Θα μάθουμε επίσης πώς μπορούμε να φτιάξουμε ένα αποθετήριο για αρχεία PCAP, καθώς και πώς μπορεί να μας φανεί χρήσιμο σε διάφορα σενάρια. Προς το παρόν σας αφήνουμε να μελετήσετε μόνοι σας και να κάνετε τις ασκήσεις που σας δώσαμε. Είμαστε σίγουρα ότι θα απολαύσετε όλα όσα έπονται ;)



Skill: Beginner

Tags: education, hacking, challenges

Καλή χακαδημαϊκή χρονιά,

από όποτε θέλετε!

Ένα ερώτημα που απασχολεί πολλούς νέους (κάθε ηλικίας), είναι αν μπορεί ποτέ κάποιος να διδαχθεί το χάκινγκ. Μπορεί δηλαδή ποτέ να υπάρξει ένα σχολείο για χάκερς; Η απάντηση έχει σχέση με το περίφημο σχολείο Hogwarts School of Witchcraft and Wizardry.

by Thiseas

Ναι, καλά το διαβάσατε. Είναι η περίφημη σχολή μαγείας στην οποία πήγε ο Harry Potter. Αυτή είναι ακριβώς που ψάχνουμε. Για να πας όμως εκεί, πρέπει να καταφέρεις να ανακαλύψεις την κρυφή αποβάθρα. Υπάρχει άραγε τέτοιος τόπος; Κι αν υπάρχει, μπορεί κάποιος να τον ανακαλύψει; Ίσως και να μπορεί!

Ο αρχαίος Έλληνας Σωκράτης είχε πει το εξής σοφό:

> *Πριν ξεκινήσουμε μια συζήτηση καλό θα είναι να ορίσουμε τις έννοιες για τις οποίες θα συζητήσουμε, διαφορετικά θα νομίζει ο καθένας ό,τι θέλει.*

Ας σεβαστούμε κι εμείς τη συγκεκριμένη ρήση κι ας προσπαθήσουμε να σας περιγράψουμε, πρώτα, τις βασικές έννοιες, τις οποίες θα ονομάσουμε αξιώματα και θεωρήματα, γύρω από τις οποίες θα κινηθεί το παρόν άρθρο.

Αξίωμα 1: Το πραγματικό[1] χάκινγκ δεν είναι κάτι (κακό ή παράνομο)[2].

Αξίωμα 2: Ο αληθινός χάκερ έχει πάρα πολλές γνώσεις και ιδιαίτερη ευφυία, κατά συνέπεια δεν μπορεί ο καθένας να γίνει χάκερ. Πολλοί νομίζουν ότι είναι, αλλά δεν είναι.

Αξίωμα 3: Δεν μπορεί να υπάρξει σχολή που να σε διδάξει πώς να γίνεις χάκερ.

Θεώρημα 1: Μπορεί να υπάρξει σχολή που να σε διδάξει πως να σκέφτεσαι. Μπορεί, δηλαδή, να σου δείξει το μονοπάτι προς το χάκινγκ.

Παρακαλώ, μην περιμένετε να εξηγήσουμε τα παραπάνω τρία αξιώματα. Η ερμηνεία τους θεωρείται προφανής. Επίσης, πρόκειται για αξιώματα: τα δεχόμαστε αξιωματικά :) Αυτό λοιπόν που πρέπει ν' αποδείξουμε είναι το Θεώρημα 1. Μπορεί, άραγε, να διδαχθεί ο τρόπος σκέψης του χάκερ; Κι αν ναι, πώς; Θα μπορούσε ας πούμε το χάκινγκ να έμπαινε ως μάθημα στα πανεπιστήμια της χώρας μας (κι όχι μόνο); Αν η απάντηση ήταν αρνητική, τότε αυτό το άρθρο θα τέλειωνε εδώ. Αν όμως τέλειωνε εδώ, ο αγαπητός αρχισυντάκτης θα το γύριζε πίσω στον γράφοντα πριν καν δημοσιευτεί, λέγοντας του: «Please, baby, don't waste my time!» (Σ.τ.Ε. Ή κάτι τέτοιο, τέλος πάντων.) Άρα, για να το διαβάζετε τώρα, σημαίνει ότι η απάντηση είναι καταφατική. Μένει, λοιπόν, να σας το αποδείξουμε.

Στις αρχές του 2005 είχε ξεκινήσει μια ιδέα από τον γράφοντα, περί της υλοποίησης με νέο περιεχόμενο και σε ελληνικά hacking fora των περίφημων τότε hacking challenges, που κυκλοφορούσαν σε δύο πολύ γνωστά διεθνή fora της εποχής: το hackthissite.org και το hellboundhackers.org. Η ιδέα υλοποιήθηκε γύρω στο 2006 στο τότε Ελληνικό hacking forum secfreaks.tk. Αργότερα ακολούθησαν κι άλλες υλοποιήσεις στα secfreaks.gr, s3cure.gr και p0wnbox.com. Επρόκειτο για κάποιες ασκήσεις χάκινγκ, όπου οι επιθέσεις γίνονταν σε εικονικά sites που είχε φτιάξει ο γράφοντας και σε απόλυτα ελεγχόμενο περιβάλλον. Κάτι σαν ένα hacking simulation. Φυσικά, δεν υπήρχαν μόνο sites αλλά και προγράμματα που έπρεπε να σπάσει κάποιος, κρυμμένα μηνύματα που έπρεπε να βρει κ.ο.κ.

Ήταν κάτι πρωτόγνωρο για τα τότε ελληνικά δεδομένα. Η ιδέα μάλλον άρεσε κι άρχισε να προχωρά. Παρουσιάστηκε σε συνέδρια, έγινε Open Source project κ.λπ. Την ίδια περίοδο την ανακάλυψαν κι έδειξαν ενδια-

1. Όπως καταλαβαίνετε, το μυστικό κρύβεται στην λέξη πραγματικό.
2. Η λέξεις είναι σε παρένθεση (βλ. Βασικές Αρχές Κατηγορικού Λογισμού), διότι αν αναλύσετε λογικά την πρόταση θα οδηγηθείτε στην ισοδύναμη της: Το πραγματικό χάκινγκ δεν είναι ούτε κακό, ούτε παράνομο.

φέρον κάποια φωτεινά, κατά την άποψη του γράφοντα, μυαλά της εποχής. Κάποιοι, δηλαδή, που μπορούν να διακρίνουν τι μπορεί να βρίσκεται κι έξω από το τείχος[3]. Πρόκειται για τους Βασίλη Βλάχο, καθηγητή του τμήματος Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών του ΤΕΙ Λάρισας, Κωνσταντίνο Παπαπαναγιώτου, συντονιστή της Ελληνικής ομάδας OWASP[4] (www.owasp.gr), κι Αλέξανδρο Παπανικολάου, επίσης καθηγητή στο προαναφερθέν τμήμα του ΤΕΙ Λάρισας. Αρχικά, εκείνοι κι ένας από τους δύο βασικούς developers του έργου, ο Αναστάσιος Στασινόπουλος, ξεκινήσαμε μια προσπάθεια να εισάγουμε όλο αυτό το έργο τόσο στην ακαδημαϊκή κοινότητα όσο και στη διεθνή σκηνή security. Αποφασίσαμε να ονομάσουμε το έργο Hackademic, εφόσον η πρώτη του εφαρμογή θα ήταν ως μέρος του μαθήματος Ασφάλεια & Διαχείριση Δικτύων, στο ΤΕΙ Λάρισας (βλ. εικόνα 1).



Εικόνα 1
Hackademic Challenges: Επίσημο εργαλείο του οργανισμού OWASP.

Το OWASP Hackademic Challenges είναι ένα εργαλείο που μας επιτρέπει να δοκιμάσουμε τις γνώσεις μας περί ασφάλειας πληροφοριακών συστημάτων και Διαδικτύου. Μπορούμε με αυτό να πραγματοποιήσουμε ελεγχόμενες επιθέσεις στα πλαίσια διαφόρων προκατασκευασμένων σεναρίων, προκειμένου να κατανοήσουμε καλύτερα τα εξής:

- Τον τρόπο σκέψης των επιτιθεμένων.

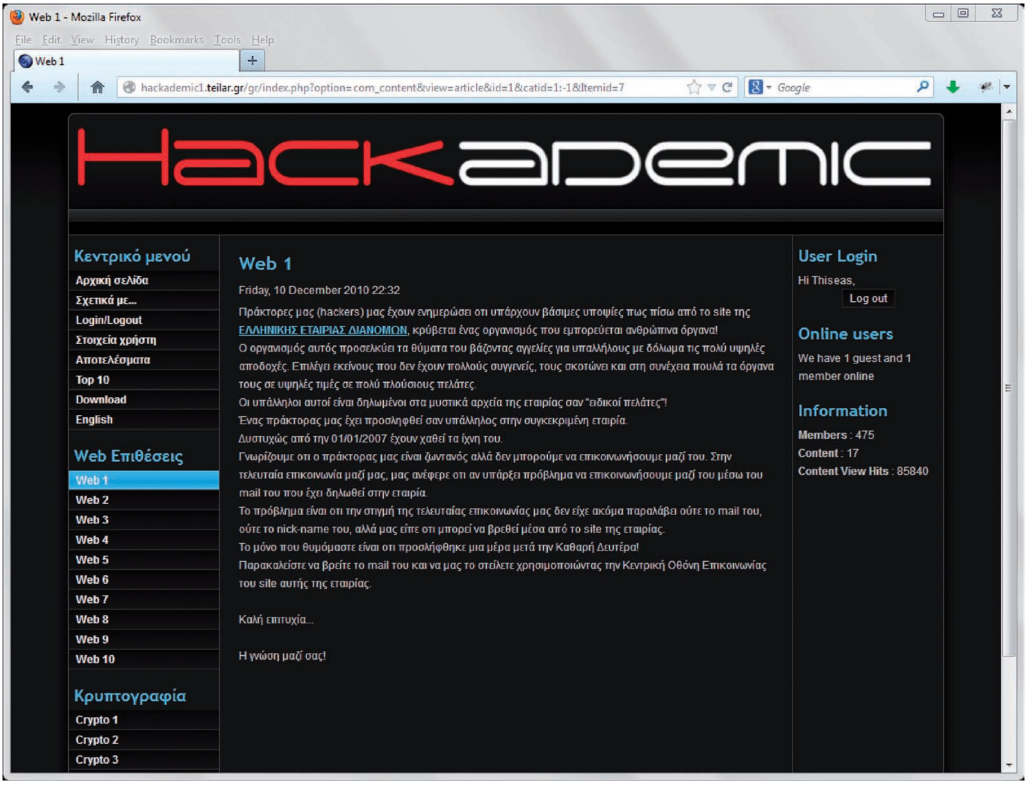
3. Βλέπε Pink Floyd, The Wall, http://el.wikipedia.org/wiki/The_Wall
4. OWASP, Open Application Security Project. Πρόκειται για τον πολύ γνωστό, παγκόσμιο, μη κερδοσκοπικό, εθελοντικό οργανισμό που ασχολείται με την ασφάλεια του λογισμικού στο Διαδίκτυο.

- Τις σχετικές μεθόδους που χρησιμοποιούνται για μια επίθεση.
- Τις συνέπειες της πλημμελούς ασφάλισης των διαδικτυακών εφαρμογών αλλά και των υποδομών που τις υποστηρίζουν.

Η υλοποίηση των σεναρίων έχει γίνει στην Ελληνική αλλά και στην Αγγλική γλώσσα και η βασική εφαρμογή βρίσκεται στους servers του ΤΕΙ Λάρισας. Μπορείτε να γραφτείτε και να παίξετε στα Ελληνικά στο `http://hackademic1.teilar.gr/gr` και στα Αγγλικά στο `http://hackademic1.teilar.gr`.

Επειδή όμως η ουσιαστική μάθηση έρχεται μέσα από την ίδια την πράξη, πάμε να παίξουμε μαζί το πρώτο σενάριο (βλ. εικόνα 2), ώστε να γίνει καλύτερα αντιληπτό για τι πράγμα μιλάμε. Αμέσως μετά θα περάσουμε σε μερικά πιο απόκρυφα στοιχεία του Hackademic, όπου θα μιλήσουμε για το τι ακριβώς βρίσκεται στα παρασκήνια.

Εικόνα 2
Web #1: Στην πρώτη άσκηση του Hackademic αναφέρεται ο στόχος και δίνεται το κίνητρο!



Παραθέτουμε πρώτα την εκφώνηση, για να την εξηγήσουμε λίγο.

> *Πράκτορές μας (hackers) μάς έχουν ενημερώσει ότι υπάρχουν βάσιμες υποψίες πως πίσω από το site της ΕΛΛΗΝΙΚΗΣ ΕΤΑΙΡΕΙΑΣ ΔΙΑΝΟΜΩΝ κρύβεται ένας οργανισμός που εμπορεύεται ανθρώπινα όργανα!*
>
> *Ο οργανισμός προσελκύει τα θύματά του βάζοντας αγγελίες για υπαλλήλους με δόλωμα τις πολύ υψηλές αποδοχές. Επιλέγει εκείνους που δεν έχουν πολλούς συγγενείς, τους δολοφονεί και στη*

συνέχεια πουλά τα όργανά τους σε υψηλές τιμές, σε πολύ πλούσιους πελάτες. Οι υπάλληλοι αυτοί είναι δηλωμένοι στα μυστικά αρχεία της εταιρείας ως «ειδικοί πελάτες».

Ένας πράκτοράς μας έχει προσληφθεί ως υπάλληλος στη συγκεκριμένη εταιρεία. Δυστυχώς, από την 01/01/2007 έχουν χαθεί τα ίχνη του. Γνωρίζουμε οτι ο πράκτορας είναι ζωντανός, αλλά δεν μπορούμε να επικοινωνήσουμε μαζί του. Στην τελευταία επικοινωνία μαζί μας, ανέφερε ότι αν υπάρξει πρόβλημα θα πρέπει να επικοινωνήσουμε μαζί του μέσω του email που έχει δηλωθεί στην εταιρεία. Το πρόβλημα είναι ότι τη στιγμή της τελευταίας επικοινωνίας δεν είχε ακόμα παραλάβει διεύθυνση email. Μας είπε όμως ότι μπορεί να βρεθεί μέσα από το site της εταιρείας.

Το μόνο που θυμόμαστε είναι οτι προσλήφθηκε μια μέρα μετά την Καθαρή Δευτέρα. Παρακαλείστε να βρείτε το email του και να μας το στείλετε, χρησιμοποιώντας την Κεντρική Οθόνη Επικοινωνίας του site της εταιρείας.

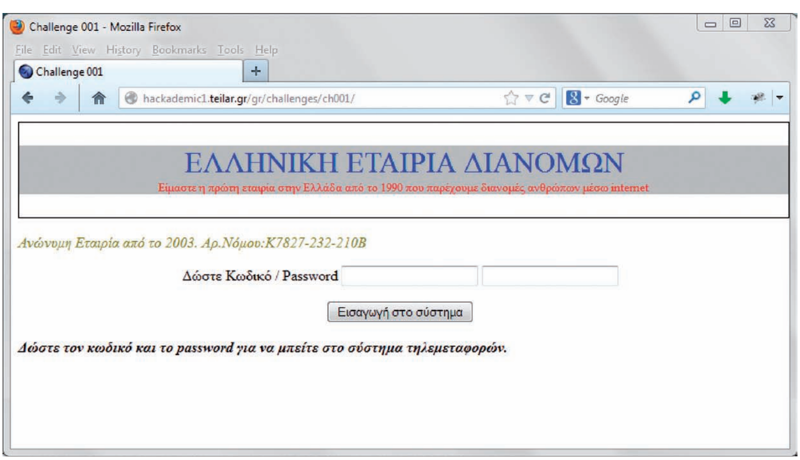
Καλή επιτυχία,
Η γνώση μαζί σας!

Διαβάζοντας την εκφώνηση, καταλαβαίνουμε τι ακριβώς ψάχνουμε να βρούμε. Επίσης, έχουμε πάντα στο μυαλό τα στοιχεία που δίνονται, διότι πιθανόν αυτά να λειτουργήσουν και σαν hints. Ψάχνουμε λοιπόν ένα συγκεκριμένο email ενός συγκεκριμένου προσώπου και το μόνο που έχουμε είναι ένα URL:

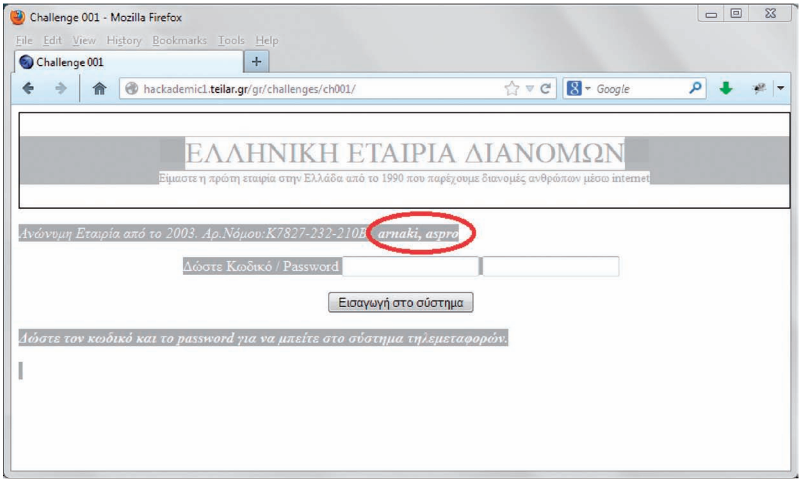
`http://hackademic1.teilar.gr/gr/challenges/ch001`

Πατώντας αυτό το URL βρισκόμαστε στην κεντρική σελίδα του προς επίθεση site (βλ. εικόνα 3).

Ως συνήθως, αρχίζουμε να περιεργαζόμαστε το site και συγκεκριμένα την αρχική αυτή σελίδα. Καταλαβαίνουμε πως ψάχνουμε τρόπο να μπούμε, δηλαδή να βρούμε κάποια credentials ή τέλος πάντων να παρακάμψουμε αυτή την ασφάλεια. Όποιος ψάξει λίγο (ανάλογα και με την εμπειρία του) δεν θα αργήσει να βρει ότι η λύση είναι ...μπροστά του, απλά δεν φαίνεται διότι είναι γραμμένη με άσπρα γράμματα σε άσπρο φόντο! Αν πατήσει [CTRL+A], τότε θα δει και τα credentials (βλ. εικόνα 4).



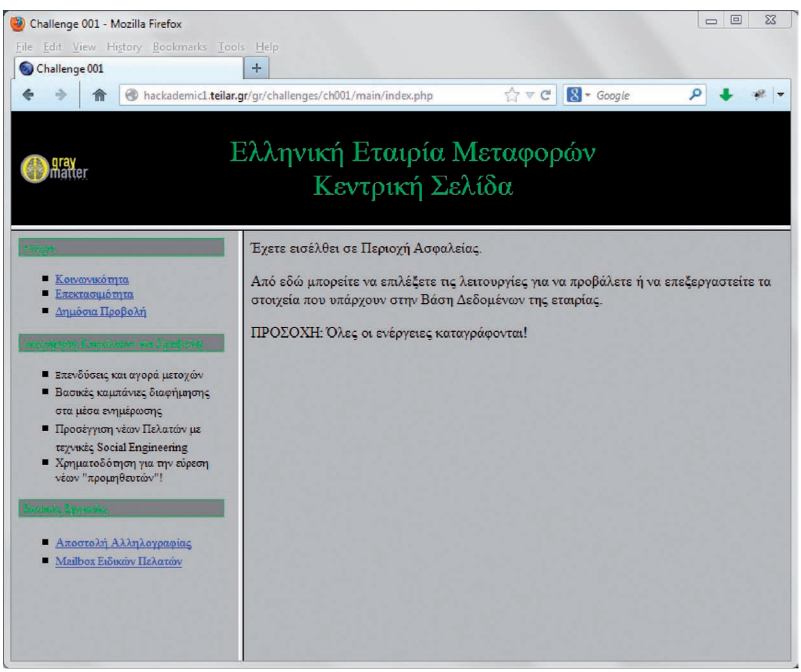
Εικόνα 3
Βρείτε τα credentials!



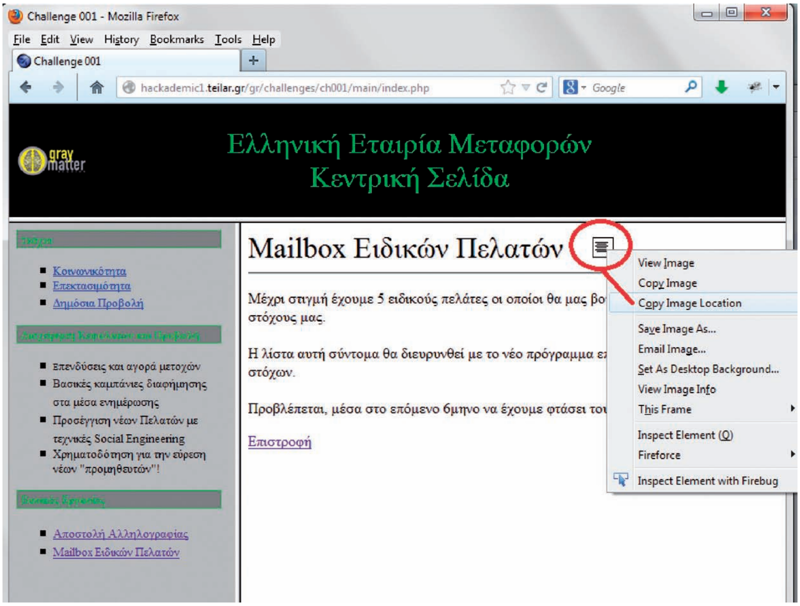
Εικόνα 4
Βρείτε τα credentials: Αρνάκι άσπρο και παχύ, της μάνας του καμάρι!

Φυσικά, παρόμοιο αποτέλεσμα θα είχατε πάρει να κάνατε View Source και βλέπατε τον κώδικα HTML της σελίδας που έχει κατέβει στον web client. Πληκτρολογώντας τα credentials μπαίνουμε, αισίως, στο κεντρικό site (βλ. εικόνα 5). Εκτός από το κλασικό ψαρωτικό μήνυμα ότι όλες οι ενέργειες μας καταγράφονται, βλέπουμε και κάποια μενού στην αριστερή στήλη που οδηγούν σε αντίστοιχες περιγραφές.

Εδώ οφείλουμε να επιδείξουμε παρατηρητικότητα και υπομονή. Πρέπει να βρούμε κάτι που να μας εμφανίσει το email του φίλου μας. Τα πιο πιθανά μέρη είναι τα δύο τελευταία μενού (στις Βασικές Εργασίες), που έχουν να κάνουν με αλληλογραφία. Για παράδειγμα, ψάχνοντας μέσα στη



Εικόνα 5
Waw, I'm in!



Εικόνα 6
Για να δούμε πού καταχωρούνται οι εικόνες…

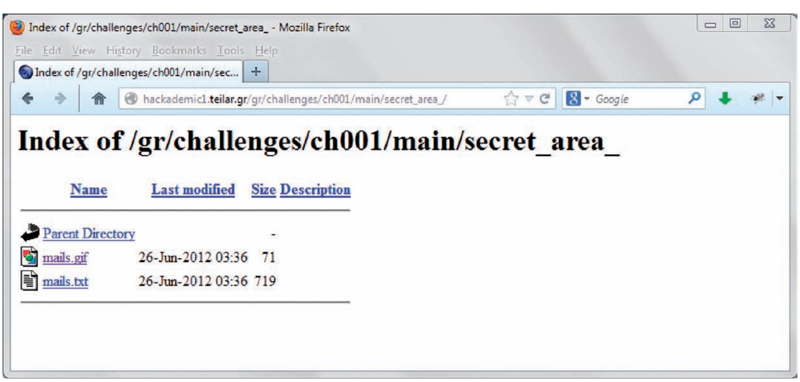
φόρμα του email των βασικών πελατών, βλέπουμε ότι το εικονίδιο που χρησιμοποιείται βρίσκεται σε έναν κατάλογο με πολύ ενδιαφέρον όνομα (βλ. εικόνα 6). Το όνομα το βρίσκουμε με δεξί κλικ επάνω στο εικονίδιο κι επιλέγοντας Copy Image Location (σε Firefox, πού αλλού;) παίρνουμε στο buffer όλο το URL πίσω από το εικονίδιο.

Αποκαλύπτουμε το URL με ένα απλό paste:

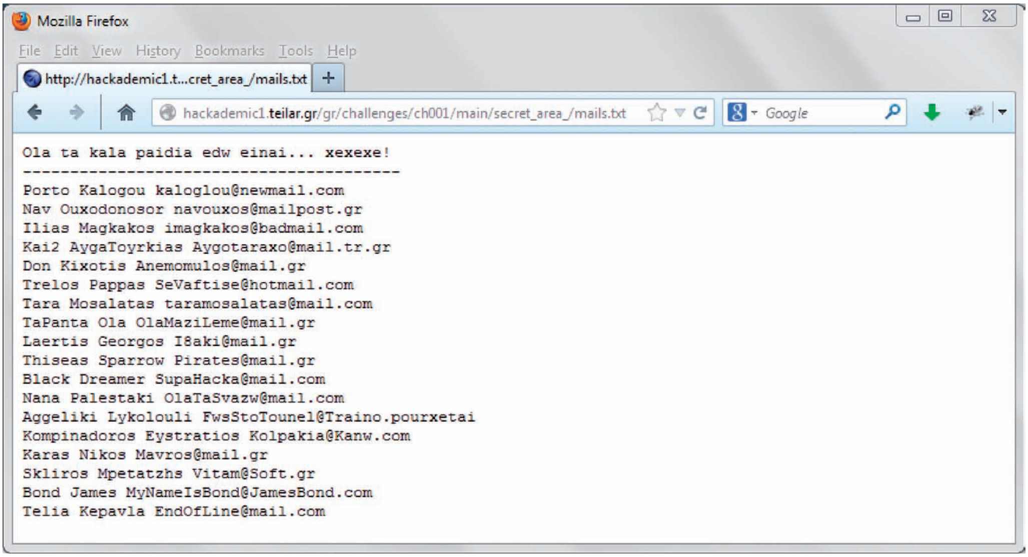
```
http://hackademic1.teilar.gr/gr/challenges/ch001/main/secret_
area_/mails.gif
```

Χμ, ενδιαφέρον, μην πείτε! Το GIFάκι βρίσκεται μέσα σε έναν κατάλογο με τ' όνομα secret_area. Ας τσεκάρουμε, λοιπόν, μήπως κάποιος αφηρημένος έχει ξεχάσει το Allow-Directory-Listing (βλ. εικόνα 7).

Βλέπω ότι, εκτός του εικονιδίου, υπάρχει και το αρχείο mails.txt. Η προκαθορισμένη συμπεριφορά των web servers όποτε ζητείται ένα αρχείο με MIME TYPE το TXT είναι να εμφανίζουν τα περιεχόμενα του. Για να δούμε, λέτε να έχουν αλλάξει αυτή τη συμπεριφορά οι admins της σελίδας;



Εικόνα 7
Νάταααα! Μήπως ψάχνουμε το αρχείο mails.txt;



Εικόνα 8
Όλα τα καλά παιδιά είν' εδώ...

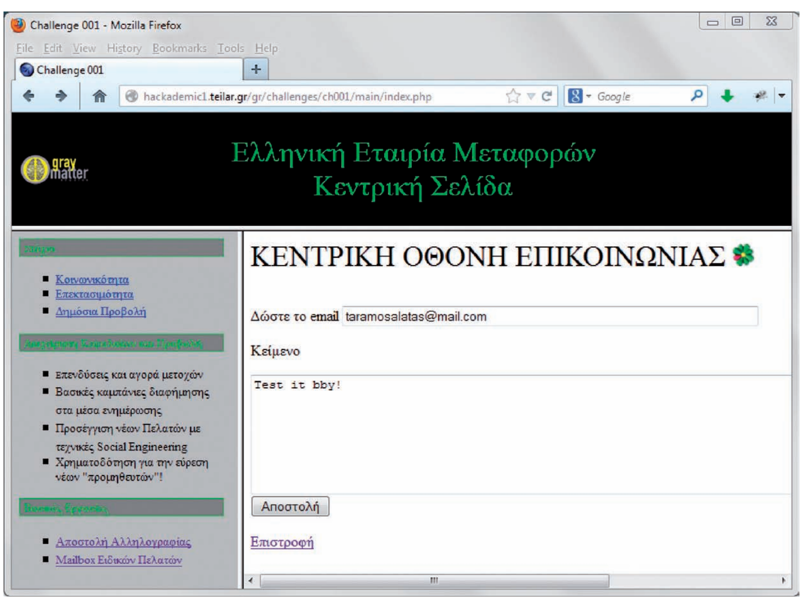
Δεν έχουμε παρά να δώσουμε, στη μπάρα διευθύνσεων του browser, το ακόλουθο URL:

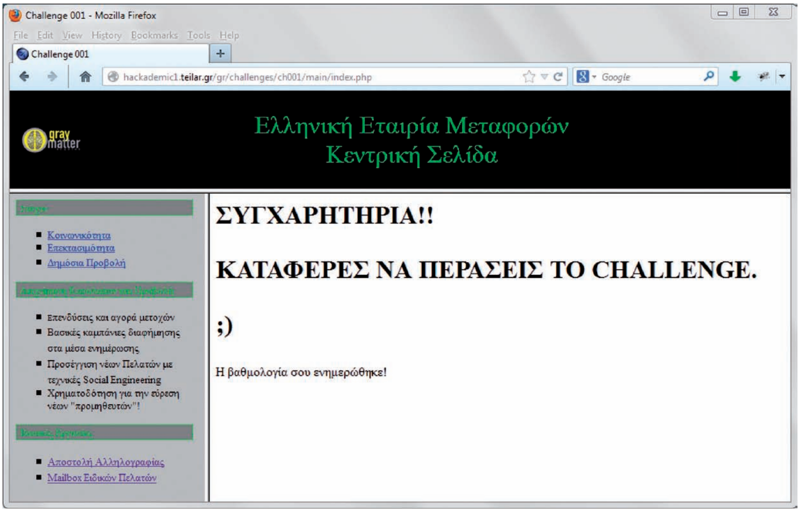
```
http://hackademic1.teilar.gr/gr/challenges/ch001/main/secret_
area_/mails.txt
```

Τα αποτελέσματα φαίνονται στην εικόνα 8 και θα λέγαμε ότι είναι εντελώς ενθαρρυντικά!

Βλέπουμε μια λίστα από emails. Είναι πολύ πιθανόν κάποιο από αυτά να είναι και του πράκτορά μας. Έχουμε δύο επιλογές: **1)** να τα δοκιμάσουμε όλα, ένα προς ένα, μέχρι να βρούμε ποιο είναι αυτό που ψάχνουμε και

Εικόνα 9
Για να δούμε, τι θα δούμε...





Εικόνα 10
Congratz, baby!

2) να χρησιμοποιήσουμε την παρατηρητικότητα μας για να βρούμε το σωστό. Αν θυμάστε, στην εκφώνηση αναφέρεται ότι ο πράκτοράς μας προσελήφθηκε μια μέρα μετά την Καθαρή Δευτέρα. Αυτό ίσως σημαίνει ότι το πρώτο υποψήφιο πρόσωπο είναι ο κύριος Tara Mosalatas, με email το taramosalatas@mail.com.

Πού όμως θα βάλουμε αυτό το email, ώστε να ενημερώσουμε την ομάδα; Αν είχαμε ψάξει καλά το site, θα βλέπαμε ότι στην επιλογή Αποστολή Αλληλογραφίας υπάρχει μια φόρμα στην οποία συμπληρώνει κάποιος email και κείμενο, το οποίο μπορεί να στείλει (βλ. εικόνα 9).

Βάζοντας λοιπόν τα στοιχεία που βρήκαμε στη φόρμα, κάνουμε αποστολή και μετά απ' αυτό νομίζουμε ότι δεν χρειάζεται κάτι άλλο (βλ. εικόνα 10).

Μόλις λύσαμε το πρώτο challenge του Hackademic!

Η θεωρία

Η επιλογή ενός challenge, η σειρά του καθώς κι ο τρόπος που λύνεται, δεν είναι καθόλου τυχαία. Υπάρχουν συγκεκριμένοι κανόνες που πρέπει κάθε challenge να ακολουθεί:

1. Να έχει **σενάριο** (μύθο).
2. Να έχει συγκεκριμένο **γνωσιολογικό στόχο** (ή στόχους).
3. Δεν πρέπει να λύνεται με διαφορετικό τρόπο από αυτόν για τον οποίο φτιάχτηκε.
4. **Να μην γίνεται γνωστό** από την εκφώνηση του σεναρίου το είδος της γνώσης που απαιτείται από πλευράς του ασκούμενου (αν, π.χ., το challenge λύνεται με SQL Injection, CSRF κ.ο.κ.). Υπάρχουν κάποιες εξαιρέσεις όπου ο κανόνας παραβιάζεται, αλλά μόνο και μόνο για χάρη των επόμενων challenges.
5. Πρέπει να υπάρχει **κλιμάκωση** και ένα **σενάριο απόδοσης γνώσεων** που υποβόσκει πίσω από όλο το σετ των ασκήσεων. Το σενάριο πρέπει να συμβαδίζει με τη λογική αλλά και με μια ενιαία στρατηγική μάθησης.

6. Οι πρώτες ασκήσεις καλό είναι να 'ναι **εύκολες και βατές** σε σύγκριση με τις επόμενες, ώστε ο μαθητευόμενος να μπει στο κλίμα πιο ομαλά και κλιμακωτά.

7. Κάθε άσκηση θα πρέπει να δίνει στον παίκτη **συγκεκριμένο στόχο** και να έχει **διακριτό τέλος,** ώστε να μην αναρωτιέται αν την έχει λύσει ή αν πρέπει να κάνει κάτι άλλο.

Το Hackdemic περιλαμβάνει δέκα ασκήσεις web. Αποτελεί πλέον θέμα χρόνου, διάθεσης και συνεισφοράς, ώστε οι 10 ασκήσεις να γίνουν 20, 30 κ.ο.κ. Κάθε άσκηση, λοιπόν, κρύβει κάποιον συγκεκριμένο στόχο μάθησης. Δυστυχώς, δεν μπορούμε εδώ να σας αναφέρουμε και τους 10 αυτούς τρόπους διότι έτσι θα παραβιάσουμε τον κανόνα 4 (παραπάνω) για όλη την κοινότητα. Μπορούμε όμως να σας αναφέρουμε τι κρύβεται πίσω από την άσκηση 1 που μόλις κάναμε.

Web Challenge 1. Ο στόχος είναι να εισαγάγει το χρήστη στις πρακτικές της αναγνώρισης και της παρατήρησης (search & observation). Ο παίκτης θα πρέπει να διερευνήσει σπιθαμή προς σπιθαμή τον υπό εξέταση χώρο. Οποιαδήποτε πληροφορία βρίσκει, ίσως χρησιμοποιηθεί ως βάση για τα επόμενα βήματα. Δύο πράγματα θα δοκιμαστούν εδώ.

1. Η ικανότητα του παίκτη για έλεγχο του πηγαίου κώδικα της σελίδας. Τα username και password βρίσκονται μπροστά στα μάτια του, αλλά η γραμματοσειρά και το χρώμα του φόντου είναι τα ίδια! Για να τα δει, ο χρήστης θα πρέπει να εξετάσει τον πηγαίο κώδικα της σελίδας. Εναλλακτικά, μπορεί να πατήσει [Ctrl+A], ενέργεια που θα προκαλέσει την αλλαγή του χρώματος του φόντου.

2. Η δεύτερη μέθοδος είναι να βρει μια εικόνα που βρίσκεται σε έναν κρυφό κατάλογο. Αυτός ο κατάλογος επιτρέπει directory listing. Ένα πολύ σημαντικό αρχείο βρίσκεται εκεί. Ο χρήστης θα πρέπει να το επιλέξει, ώστε να δει τα περιεχόμενά του. Επίσης από την αναζήτηση κι αναγνώριση του site πρέπει να ξέρει με ποιόν τρόπο θα στείλει τα αποτελέσματα που θα βρει.

Η πράξη

Για την υλοποίηση της online εκδοχής του OWASP Hackademic Challenges, χρησιμοποιήθηκαν αποκλειστικά και μόνο εργαλεία ανοιχτού κώδικα.

- Ένας server (http://teilar.gr)
  Fedora 14, http://fedoraproject.org
- Ένα σύστημα διαχείρισης περιεχομένων (CMS)
  Joomla, http://joomla.org
- Ένα σύστημα διαχείρισης βάσεων δεδομένων (ΣΔΒΔ)
  MySQL, http://mysql.com

Μπορώ να το τρέξω στο PC μου;

Ναι! Μπορείτε να κατεβάσετε τον κώδικα του hackademic και να το τρέξετε τοπικά, χωρίς να απαιτείται να έχετε εγκατεστημένο το Joomla. Για την τοπική εκτέλεση του OWASP Hackademic Challenges, στο σύστημα θα πρέπει να υπάρχει εγκατεστημένος (κι ενεργός) ο **Apache Server,** καθώς και η **PHP**.

Στο Fedora, για την εγκατάσταση του Apache και της PHP δώστε

```
[root@hackademic ~]# yum -y install httpd php
```

Ενεργοποιήστε τον Apache με

```
[root@hackademic ~]#  /sbin/service httpd start
```

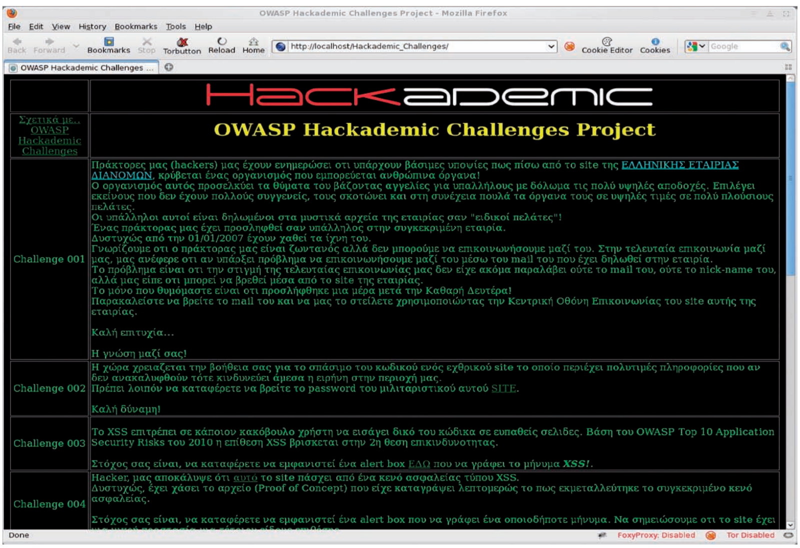
Μπορείτε να κατεβάσετε τον κώδικα από τη σχετική σελίδα του Google Code:

https://code.google.com/p/owasp-hackademic-challenges

Αποσυμπιέστε και αντιγράψτε τον φακέλο Hackademic_Challenges στον κατάλογο /var/www/html (με τα ανάλογα δικαιώματα). Τέλος, εισάγετε στον αγαπημένο σας web browser τη διεύθυνση

http://localhost/Hackademic_Challenges

Θα δείτε τα challenges με τη μορφή που φαίνονται στην εικόνα 11.



Εικόνα 11
Τα Hackademic challenges, σε τοπικό υπολογιστή.

Αποδείξαμε το Θεώρημα 1;

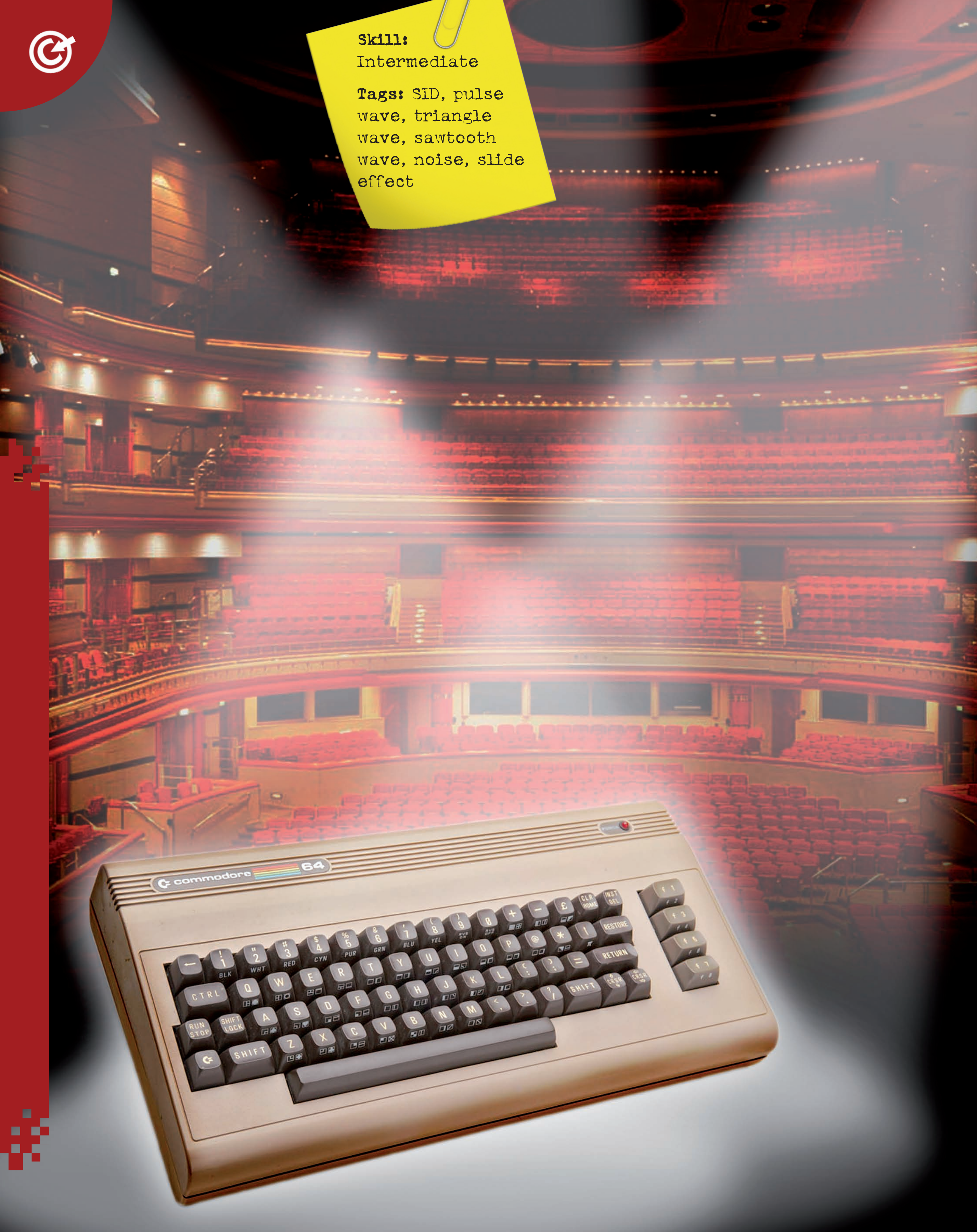
Το Hackademic έγινε γρήγορα γνωστό και σύντομα υιοθετήθηκε ως παγκόσμιο επίσημο project του OWASP. Χρησιμοποιήθηκε και στο πανευρωπαϊκό συνέδριο OWASP AppSec Europe 2011, ως μέσο για τη δωρεάν εισαγωγή και παρακολούθηση του συνεδρίου. Έχει παρουσιαστεί με μεγάλη επιτυχία σε πανελλήνια συνέδρια όπως τα...

- Conference / OSS 2011 National Technical University
  http://conferences.ellak.gr/2011
- Conference Fosscomm 2011 University of Patras
  http://patras.fosscomm.gr/schedule
- Conference OpenFest 2011 festival FS / RA in TEI Piraeus
- ΕΛΛΑΚ & Hackademic (PDF)
  http://bit.ly/ellakhackademic

Εμείς πιστεύουμε ότι το θεώρημα το έχουμε αποδείξει. Την επαλήθευση όμως θα την κάνετε εσείς. Η γνώση μαζί σας!

Skill: Intermediate

Tags: SID, pulse wave, triangle wave, sawtooth wave, noise, slide effect



Ήχος από το παρελθόν!

Arduino Sound Master

part 4/όσο χρειαστεί

Αρκετά με τα πιάνα, τις κιθάρες και τις τρομπέτες! Ο κόσμος θέλει να ακούσει «βρώμικους» ηλεκτρονικούς ήχους, γεμάτους artifacts και γενικά άκρως νοσταλγικούς. Τι θα λέγατε αν γνωρίζαμε το τσιπάκι ήχου που συνόδευε έναν από τους πιο πετυχημένους υπολογιστές όλων των εποχών; Αναφερόμαστε στον Commodore 64, του οποίου τα παιχνίδια και κυρίως η μουσική τους άφησαν εποχή!

by Spir@l Evolut10n

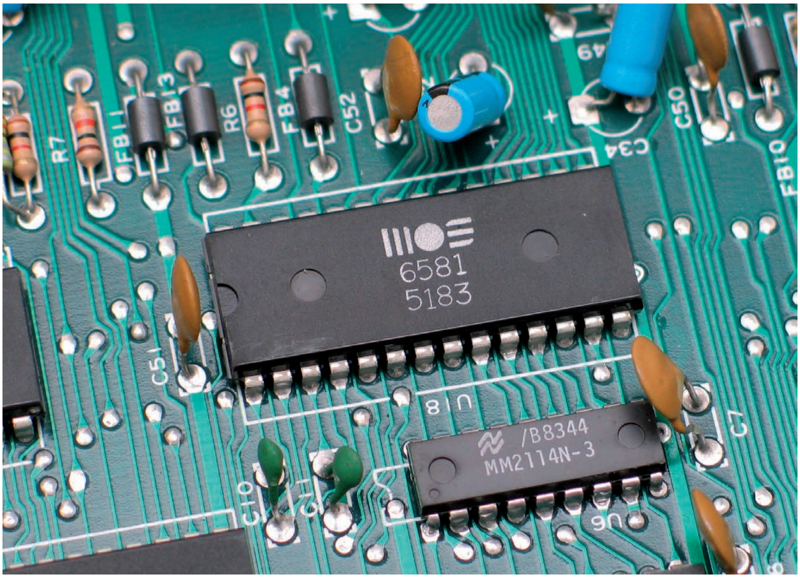
Ο ήχος ενός απλού αρμονικού κύματος μοιάζει αρκετά με τους φυσικούς ήχους και, τέλος πάντων, ακούγεται ευχάριστα. Εμείς όμως ξεκινήσαμε την εξερεύνηση των τεχνικών σύνθεσης ήχου για να γνωρίσουμε μερικές ευρηματικές προγραμματιστικές τεχνικές και, κυρίως, για να ικανοποιήσουμε τη νοσταλγική μας διάθεση. Στην αρχή αυτής της σειράς, λοιπόν, αναφερθήκαμε στους ακατέργαστους ήχους που παρήγαγαν οι παλιές παιχνιδομηχανές. Δεν μιλάμε για απλοϊκά μπλιπ και μπλοπ, αλλά για τις καταιγιστικές μελωδίες, με το χαρακτηριστικό «ηλεκτρονικό θόρυβο» που ακούγαμε από παιχνίδια στην Amiga, στον Commodore και στα παλιά arcades. Μη νομίζετε ότι η νοσταλγική διάθεση απέναντι σε αυτούς τους ήχους αποτελεί αποκλειστικά δική μας τρέλα. Με το θέμα έχουν ασχοληθεί πολλοί προγραμματιστές αλλά και μουσικοί, που ανήκουν στη λεγόμενη demo scene. Σε αυτή την κοινότητα ανήκουν όσοι γράφουν μικροσκοπικά προγράμματα σε Assembly (και *σπανίως* σε C), τα οποία ωστόσο υλοποιούν εξαιρετικά σύνθετα εφέ. Ένα τμήμα αυτής της σκηνής ασχολείται με τη σύνθεση μουσικής και τα κομμάτια τους είναι γνωστά σαν chiptunes. Παρεμπιπτόντως, με τέτοια μουσική είναι ντυμένα και τα διάφορα key-generators.

Ο βασιλιάς του ήχου

Μερικοί παλιοί υπολογιστές —τουλάχιστον οι πιο εξελιγμένοι— ήταν εξοπλισμένοι με κάποιο ειδικό τσιπάκι το οποίο αναλάμβανε την παραγωγή του ήχου. Μην φαντάζεστε τα υπερσύγχρονα ολοκληρωμένα κυκλώματα που διαθέτουν οι σημερινές κάρτες ήχου. Τα τσιπάκια εκείνης της εποχής προσέφεραν στοιχειώδεις λειτουργίες, με βασικότερη την εύκολη σύνθεση μερικών κυμάτων. Ουσιαστικά, τα εν λόγω τσιπάκια απάλλασσαν τον επεξεργαστή από ορισμένες επαναλαμβανόμενες εργασίες (υπολογισμό νέων δειγμάτων, εφαρμογή περιγράμματος έντασης κ.ά.) και με αυτόν τον τρόπο βοηθούσαν στη συνολική επιτάχυνση του συστήματος. Ωστόσο, σε σχέση με τη μουσική καθεαυτή, τα παλιά τσιπάκια ήχου προσέφεραν ελάχιστες δυνατότητες. Για να συνθέσουν τις διάφορες μελωδίες οι προγραμματιστές εξαντλούσαν κάθε φορά τα όρια της ευρηματικότητάς τους και συνδύαζαν τα ελάχιστα εφόδια με έξυπνους και πρωτότυπους τρόπους. Βέβαια, οι περιορισμοί του hardware άφηναν πάντα τη σφραγίδα τους στον ήχο και οι φανατικοί του είδους μπορούσαν να διακρίνουν εύκολα αν κάποιο μουσικό κομμάτι προερχόταν από το ένα ή το άλλο τσιπάκι. Ένας από τους πρώτους υπολογιστές που διέθετε εξειδικευμένο hardware για τον ήχο ήταν ο Commodore 64 και το σχετικό ολοκληρωμένο κύκλωμα ήταν το SID[1] 6581/8580. Ε, λοιπόν, η τεράστια επιτυχία του εν λόγω υπολογιστή και η πληθώρα των αντίστοιχων παιχνιδιών, έκαναν το συγκεκριμένο τσιπ ιδιαίτερα δημοφιλές. Έτσι, οι περισσότεροι μουσικοί και προγραμματιστές που ασχολούνται με τα chiptunes, γράφουν μουσική για το SID! Αν κάνετε μια γρήγορη αναζήτηση στο Internet θα διαπιστώσετε ότι υπάρχουν δεκάδες SID emulators, ενώ το YouTube βρίθει από μουσικά κομμάτια γραμμένα για το SID (μπορείτε να πάρετε μια γεύση από εδώ: http://bit.ly/spino2006).

Όπως υποψιάζεστε, αν και δεν σκοπεύουμε να γράψουμε έναν (ακόμα) εξομοιωτή του SID, θα μιμηθούμε αρκετές από τις λειτουργίες του συγκεκριμένου τσιπ. Όμως, τι ακριβώς μπορούσε να κάνει το SID; Να πούμε κατ' αρχάς ότι διέθετε τρία κανάλια ήχου. Καθένα απ' αυτά μπορούσε να παράγει τριγωνικά, πριονωτά και τετραγωνικά κύματα, ή θόρυβο[2]. Για την

1. Sound Interface Device
2. Το SID δεν μπορούσε να παράγει αρμονικά (ημιτονοειδή) κύματα.



Εικόνα 1
Το SID 6581, πάνω στην πλακέτα ενός Commodore 64! Πρόκειται για το τσιπάκι ήχου γύρω από το οποίο αναπτύχθηκε μια ολόκληρη κοινότητα μουσικών και προγραμματιστών.

εξομοίωση του θορύβου το SID διέθετε μια μονάδα παραγωγής ψευδοτυχαίων αριθμών, ενώ ήταν εξοπλισμένο και με τρεις μηχανισμούς ADSR, για την εφαρμογή ενός περιγράμματος έντασης (volume envelope) σε κάθε κανάλι. Το τσιπάκι δεν ήταν αμιγώς ψηφιακό. Ήταν αυτό που λέμε mixed-signal κι ενσωμάτωνε κι ορισμένα αναλογικά συστήματα (κυρίως φίλτρα). Κάπου εδώ όμως θα σταματήσουμε. Μια αναλυτική παράθεση όλων των υποσυστημάτων του SID θα ήταν κουραστική και μάλλον περιττή.

Πρωτόγονα κύματα

Όπως αναφέραμε, το SID παρείχε βασικές λειτουργίες και η σύνθεση κάθε μελωδίας επαφίετο στις ικανότητες του εκάστοτε προγραμματιστή. Σε αυτό το άρθρο, λοιπόν, θα γνωρίσουμε τη χροιά των βασικών ήχων που παρήγαγε το διάσημο τσιπ. Για το σκοπό αυτό θα φτιάξουμε ένα πρόγραμμα, το οποίο θα παράγει τριγωνικά, τετραγωνικά και πριονωτά κύματα, όπως επίσης και θόρυβο! Το πρόγραμμά μας θα βασίζεται στο γνώριμο κύκλωμα με το μικρό πληκτρολόγιο και τα δύο ποτενσιόμετρα. Το ένα ποτενσιόμετρο θα χρησιμοποιείται για την επιλογή της μορφής των παραγόμενων κυμάτων. Το άλλο ποτενσιόμετρο θα έχει διπλό ρόλο: Όταν έχουμε επιλέξει τετραγωνικά κύματα θα ρυθμίζει το duty cycle, ενώ όταν έχουμε επιλέξει το θόρυβο θα ρυθμίζει την ποιότητά του!

Τα κύματα που θα παράγει το πρόγραμμά μας θα έχουν διαφορετικό ήχο μεταξύ τους — αλλά και σε σχέση με τα ημιτονοειδή. Μάλιστα, αν τα ακούσουμε προσεκτικά θα διαπιστώσουμε ότι ο ήχος τους είναι πλούσιος, μοιάζει να απαρτίζεται από την πρόσθεση πολλών διαφορετικών αρμονικών κυμάτων και χαρακτηρίζεται απ' αυτόν τον ηλεκτρονικό θόρυβο που συνόδευε τον ήχο των παλιών παιχνιδιών! Το πρόγραμμα θα μάς φέρει σε επαφή και με ένα από τα εφέ που είχαν επινοήσει οι προγραμματιστές της εποχής του SID. Αναφερόμαστε στο λεγόμενο slide. Το συγκεκριμένο εφέ εφαρμόζεται μόνο σε τετραγωνικά κύματα και προβλέπει την παλινδρόμηση του duty cycle μεταξύ δύο τιμών. Ο ήχος που προκύπτει από τις

διαδοχικές μεταβολές στο duty cycle είναι δύσκολο να περιγραφεί, αλλά είμαστε βέβαιοι ότι τον έχετε ακούσει και θα σας φανεί γνώριμος. Η χρήση του slide ήταν πολύ διαδεδομένη στη μουσική των παλιών παιχνιδιών και γι' αυτόν το λόγο χρησιμοποιείται πολύ συχνά και στα chiptunes. Με τη βοήθεια του προγράμματος θα μπορέσουμε να εξομοιώσουμε και να ακούσουμε τον ήχο του slide, πατώντας το κουμπί μιας νότας και μεταβάλλοντας χειροκίνητα το duty cycle. Τέλος, αξίζει να σημειώσουμε ότι ο θόρυβος ήταν επίσης διαδεδομένος στα παλιά παιχνίδια. Οι προγραμματιστές τον αξιοποιούσαν για τον ήχο των καυσαερίων ενός σκάφους, για τον ήχο ορισμένων διαστημικών όπλων και πάει λέγοντας.

Όλα είναι τρίγωνα!

Σκεφτείτε την περίπτωση που πρέπει να σχηματίσουμε ένα τριγωνικό κύμα, με την τεχνική DDS και αξιοποιώντας ένα κανάλι PWM. Αρχικά, κάθε δείγμα που θα παράγουμε θα είναι *λίγο* μεγαλύτερο από το αμέσως προηγούμενο. Κάποια στιγμή, ωστόσο, θα φτάσουμε στη μέγιστη τιμή που μπορούν να πάρουν τα δείγματα. Ακριβώς τότε, το αμέσως επόμενο δείγμα θα είναι *λίγο* μικρότερο και, γενικότερα, τα δείγματα θα υιοθετήσουν πλέον μια πτωτική πορεία. Αυτό θα ισχύει μέχρι να φτάσουμε στην ελάχιστη τιμή που μπορεί να πάρει ένα δείγμα, οπότε και θα ξεκινήσει μια νέα ανοδική πορεία και πάει λέγοντας. Εάν το σύστημα PWM είχε ανάλυση 4 bits, η ανοδική πορεία των δειγμάτων θα περιελάμβανε το πολύ 16 βήματα (ή σκαλοπάτια, αν το προτιμάτε) και το ίδιο θα ίσχυε για την πτωτική πορεία. Αν το PWM είχε ανάλυση 2 bits, τότε κάθε σκέλος του τριγώνου θα διέθετε το πολύ 4 σκαλοπάτια και θα ήταν μάλλον χοντροκομμένο. Μπορείτε να φανταστείτε τι θα συνέβαινε αν το σύστημα PWM είχε ακρίβεια ενός bit; Κάθε τρίγωνο θα σχηματιζόταν από ένα ακριβώς σκαλοπάτι. Με άλλα λόγια, το τριγωνικό κύμα θα είχε εκφυλιστεί πλέον σε τετραγωνικό. Τα τετραγωνικά κύματα, λοιπόν, αποτελούν τριγωνικά κύματα ανάλυσης ενός bit! Αν αυτή η περιγραφή δεν σας έπεισε, ρίξτε μια ματιά και στην εικόνα 2.

Ο κώδικας

Κατά τα γνωστά, πριν ξεκινήσουμε την παρουσίαση του κώδικα να πούμε ότι μπορείτε να τον κατεβάσετε από το

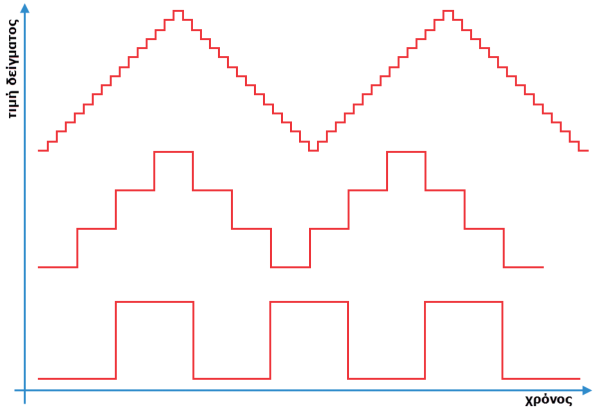
`http://bit.ly/dh019sidpiano`

Σε αυτή την έκδοση του προγράμματος η ρουτίνα υπολογισμού των νέων δειγμάτων χωρίζεται σε τρία βασικά μέρη. Στο πρώτο μέρος υλοποιείται το περίγραμμα έντασης, το οποίο αργότερα εφαρμόζεται στην επιλεγμένη κυματομορφή. Στο δεύτερο μέρος ο κώδικας υπολογίζει ένα νέο δείγμα. Για αυτόν τον υπολογισμό υπάρχουν τέσσερα επιμέρους τμήματα κώδικα — ένα για κάθε τύπο κύματος. Το πρόγραμμα εκτελεί κάθε φορά ένα μόνο από τα συγκεκριμένα τμήματα, ανάλογα με την κυματομορφή που έχει επιλέξει ο χρήστης. Στο τρίτο και τελευταίο μέρος, ο κώδικας διαβάζει τα δύο ποτενσιόμετρα και ενημερώνει κατάλληλα τις μεταβλητές waveform, dutycycle και noisequality. Στη συνέχεια θα ρίξουμε μια ματιά στα σχετικά αποσπάσματα, προσπερνώντας τους υπολογισμούς για το

περίγραμμα έντασης, αφού τους έχουμε εξετάσει προ πολλού (βλ. σχετικό άρθρο στο deltaHacker 016).

```
case 0: // -------- noise
  if (phase.int_val > noisequality) {
    phase.int_val = 0;
    prng();
    prng();
    OCR2A = rnd_num.bytes[0];
  }
  break;
```

Στην περίπτωση που έχει επιλεγεί ο θόρυβος, το νέο δείγμα προέρχεται από μια πηγή ψευδοτυχαίων αριθμών. Το πρόγραμμά μας περιλαμβάνει γι' αυτό το λόγο έναν LFSR, του οποίου την υλοποίηση εξετάσαμε στο τεύχος 018. Το ενδιαφέρον εδώ εντοπίζεται στον έλεγχο της πρώτης γραμμής. Η έκβαση αυτού του ελέγχου καθορίζει το αν θα υπολογιστεί νέο δείγμα για τον θόρυβο ή όχι. Ο έλεγχος περιλαμβάνει τη σύγκριση των phase και noisequality. Η μεταβλητή phase αυξάνεται κάθε φορά που εκτελείται η ρουτίνα υπολογισμού δειγμάτων. Βέβαια, στην περίπτωση του θορύβου, η έννοια της φάσης δεν έχει κανένα νόημα[3]! Επομένως, η σύγκριση που πραγματοποιείται εδώ εξυπηρετεί μόνο στην εισαγωγή μιας καθυστέρησης. Όσο πιο μεγάλη είναι η τιμή της μεταβλητής noisequality, τόσο πιο πολύ καθυστερεί η παραγωγή νέων δειγμάτων κι αντίστροφα. Αναρωτιέστε σε τι εξυπηρετεί αυτή η καθυστέρηση; Όσο πιο συχνά υπολογίζονται τυχαίες τιμές —δείγματα για το θόρυβο— ο παραγόμενος ήχος μοιάζει «μεταλλικός». Από την άλλη, όσο πιο αραιά παράγονται νέα δείγματα ο θόρυβος μοιάζει πιο «ομοιόμορφος» και θυμίζει εκείνο των ...καυσαερίων. Με λίγα λόγια, η καθυστέρηση που εισάγεται από το συγκεκριμένο έλεγχο καθορίζει τη χροιά του θορύβου.

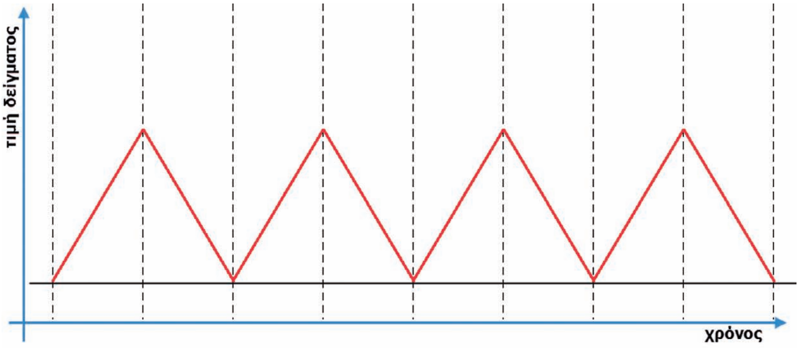


Εικόνα 2
Το πρώτο τριγωνικό κύμα είναι τετράμπιτο, αφού κάθε σκέλος περιγράφεται από 16 στάθμες (2^4). Αντίστοιχα, το δεύτερο τριγωνικό κύμα είναι των δύο bits, αφού κάθε σκέλος περιγράφεται από 4 στάθμες (2^2). Το τρίτο τριγωνικό κύμα διαθέτει δύο στάθμες σε κάθε σκέλος του (2^1) και έχει ανάλυση ένα 0bit. Βέβαια, οι φυσιολογικοί άνθρωποι θα χαρακτήριζαν το συγκεκριμένο κύμα απλά ως τετραγωνικό ;)

```
case 1: // -------- square wave
  if (sample > dutycycle) {sample = 255;}
  else  {sample = 0;}
  OCR2A = ((amp_envelope.bytes[1] * sample) >> 8);
  break;
```

Εδώ τα πράγματα είναι λίγο πιο απλά. Τα τετραγωνικά κύματα είναι περιοδικά και η έννοια της φάσης αποκτά πάλι τη γνώριμη ερμηνεία. Ένα τετραγωνικό κύμα περιγράφεται από δύο μόνο στάθμες και, κατ' επέκταση, τα δείγματα που σχηματίζουν ένα τέτοιο κύμα μπορούν να έχουν δύο

3. Ο θόρυβος δεν παρουσιάζει περιοδικότητα και κατ' επέκταση είναι αδύνατο να ορίσουμε τη φάση του.

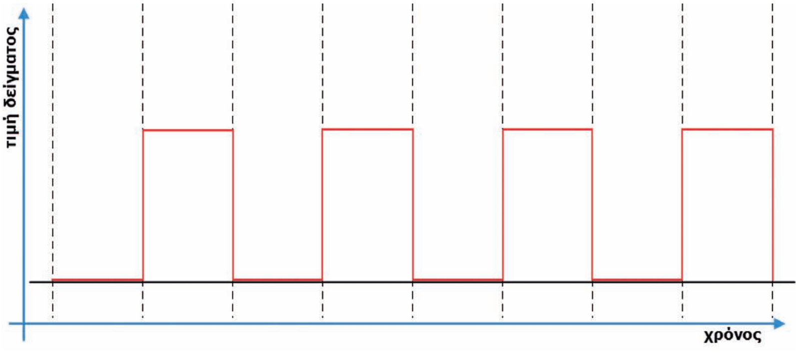


Εικόνα 3
Τα τριγωνικά κύματα χωρίζονται σε δύο σκέλη ανά περίοδο. Στο ένα σκέλος η στάθμη αυξάνεται γραμμικά ως τη μέγιστη τιμή, ενώ στο άλλο η στάθμη φθίνει (πάντα γραμμικά) ως την ελάχιστη τιμή.

μόνο τιμές (0 και 255). Πριν προχωρήσουμε με την ανάλυση του κώδικα, πρέπει να αναφέρουμε μια λεπτομέρεια. Αρχικά, η μεταβλητή sample έχει σαν τιμή το περισσότερο σημαντικό byte της μεταβλητής phase. Με άλλα λόγια, η μεταβλητή sample έχει σαν τιμή το ακέραιο μέρος της φάσης[4]. Η σύγκριση που πραγματοποιείται στην πρώτη γραμμή καθορίζει τη φάση (τη χρονική στιγμή εντός της περιόδου του κύματος), κατά την οποία η στάθμη θα μεταβεί από την τιμή 0 στην τιμή 255. Εν ολίγοις, ο συγκεκριμένος έλεγχος καθορίζει το duty cycle του παραγόμενου τετραγωνικού κύματος.

```
case 2: // -------- triangle wave
  if (sample < 128) {sample = sample * 2;}
  else  {sample = 511 - sample * 2;}
  OCR2A = ((amp_envelope.bytes[1] * sample) >> 8);
  break;
```

Στην περίπτωση των τριγωνικών κυμάτων, τα δείγματα πρέπει να αυξάνουν σταδιακά ως τη μέγιστη τιμή (255) και στη συνέχεια να φθίνουν σταδιακά ως την ελάχιστη τιμή (0). Με άλλα λόγια, για όσο χρόνο η φάση του κύματος βρίσκεται στο πρώτο μισό του εύρους της (0...127), οι τιμές των δειγμάτων που σχηματίζουν το τρίγωνο πρέπει να αυξάνονται. Αντίστοιχα, για όσο χρόνο η φάση βρίσκεται στο δεύτερο μισό του εύρους της (128...255), οι τιμές των δειγμάτων που σχηματίζουν το τρίγωνο πρέπει να φθίνουν. Το πρόγραμμα, λοιπόν, ελέγχει σε ποια περιοχή τιμών βρι-

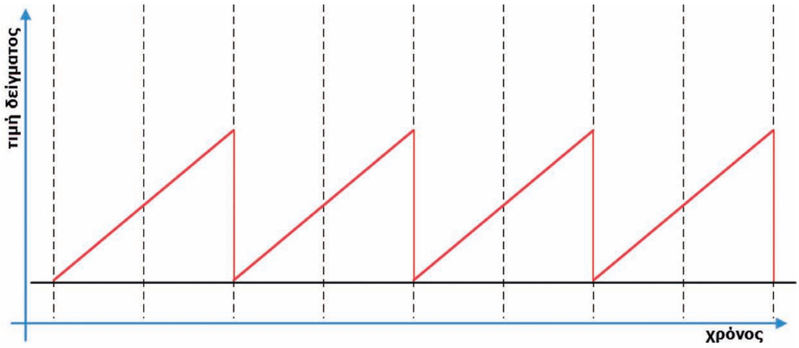


Εικόνα 4
Τα τετραγωνικά κύματα δεν χρειάζονται εισαγωγές. Τα έχουμε συναντήσει πολλές φορές και η μορφή τους είναι γνώριμη σε όλους.

4. Αυτό προκύπτει από τον τρόπο που έχουμε ορίσει τη μεταβλητή phase. Όπως θα θυμάσαστε, η συγκεκριμένη μεταβλητή αντιμετωπίζεται σαν ένας αριθμός σταθερής υποδιαστολής 8 προς 8 (δείτε το σχετικό άρθρο στο deltaHacker 015).

σκεται η φάση και κάνει τους ανάλογους υπολογισμούς. Εδώ χρειάζεται λίγη προσοχή, καθώς η μεταβλητή sample αλλάζει ρόλο! Όπως είπαμε, η εν λόγω μεταβλητή φέρει αρχικά το ακέραιο μέρος της φάσης. Όταν όμως ολοκληρωθεί ο έλεγχος της φάσης, η μεταβλητή sample χρησιμοποιείται γι' αυτό που δηλώνει το όνομά της. Η τιμή του δείγματος για το «ανοδικό» σκέλος του τριγωνικού κύματος προκύπτει διπλασιάζοντας την τιμή της φάσης. Έτσι, οι τιμές των δειγμάτων αυξάνουν σταδιακά από την τιμή 0*2 έως την τιμή 127*2. Στο άλλο σκέλος του τριγωνικού κύματος, η τιμή του δείγματος προκύπτει διπλασιάζοντας πάλι τη φάση και το αποτέλεσμα αφαιρείται από την τιμή 256. Έτσι, οι τιμές των δειγμάτων φθίνουν σταδιακά από την τιμή 511–128*2 έως την τιμή 511–255*2.

```
case 3: // -------- sawtooth wave
  OCR2A = ((amp_envelope.bytes[1] * sample) >> 8);
  break;
```



Εικόνα 5
Σε ένα πριονωτό κύμα, η στάθμη αυξάνεται γραμμικά ως τη μέγιστη τιμή. Ακριβώς τότε, η στάθμη επιστρέφει στην ελάχιστη τιμή και ξεκινά πάλι μια γραμμική αύξηση.

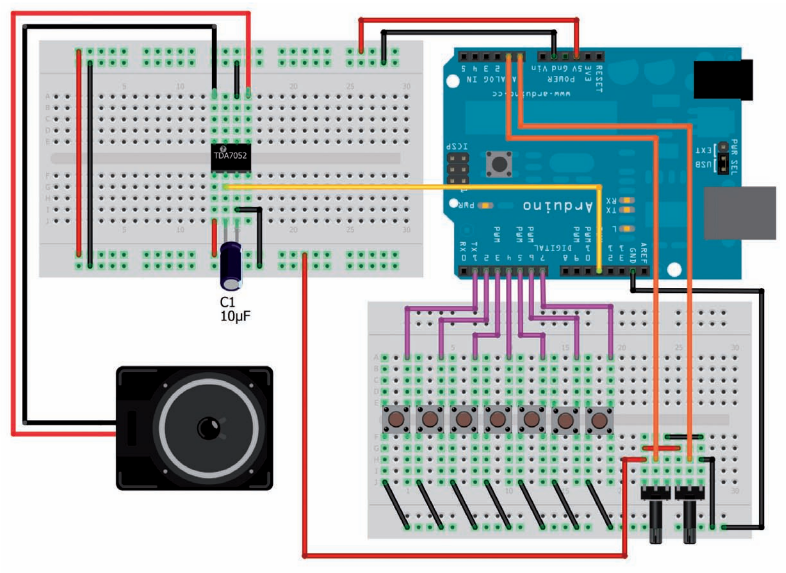
Τα πριονωτά κύματα σχηματίζονται με τον πιο εύκολο τρόπο. Το δείγμα προκύπτει απλά και μόνο από τη φάση του κύματος. Βλέπετε, η στάθμη ενός τέτοιου κύματος αυξάνει σταδιακά ως τη μέγιστη τιμή και στη συνέχει επιστρέφει στην ελάχιστη, για να αρχίσει και πάλι να αυξάνεται. Επομένως, τα αντίστοιχα δείγματα ακολουθούν την ίδια πορεία με τη φάση.

```
if (sample_cnt == 15) {
  potvalue = analogRead(A0);
  waveform = potvalue >> 8;
}
```

Εδώ πραγματοποιείται η επιλογή του κύματος, σύμφωνα με την τιμή του ενός ποτενσιόμετρου. Το κύκλωμα ADC του Arduino επιστρέφει τιμές των 10 bits (0...1023). Για τον ορισμό της μεταβλητής waveform, η τιμή που επιστρέφει ο ADC ολισθαίνει κατά 8 bits προς τα δεξιά. Έτσι, η τιμή που λαμβάνεται από τον ADC διαιρείται με το 256 και, τελικά, η τιμή της waveform κυμαίνεται στο διάστημα [0...3]. Με αυτό το τέχνασμα, το εύρος των τιμών που μπορούμε να διαβάσουμε από το ποτενσιόμετρο διαιρείται σε τέσσερεις περιοχές. Καθεμία από αυτές αντιστοιχεί σε έναν τύπο κύματος.

```
// read duty cycle pot
if (sample_cnt == 30) {
```

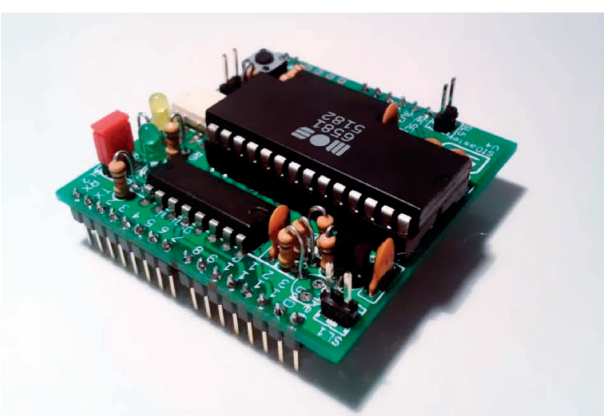
Εικόνα 6
Το (γνώριμο) κύκλωμα για το οποίο προορίζεται το πρόγραμμά μας. Αν θέλετε να απολαύσετε τους παλιομοδίτικους ήχους σε χορταστικές εντάσεις και με φουλ παραμόρφωση, μπορείτε να συνδέσετε την έξοδο του κυκλώματος απευθείας πάνω στον ενισχυτή σας ή σε κάποιο ζευγάρι ενεργών ηχείων. Η έξοδος του κυκλώματος δίνεται από το κίτρινο καλώδιο, αλλά εδώ χρειάζεται λίγη προσοχή! Για να μη δημιουργήσετε πρόβλημα στον ενισχυτή, συνδέστε ανάμεσα στο κίτρινο καλώδιο και την είσοδο του ενισχυτή έναν ηλεκτρολυτικό πυκνωτή 1μF / 16V.

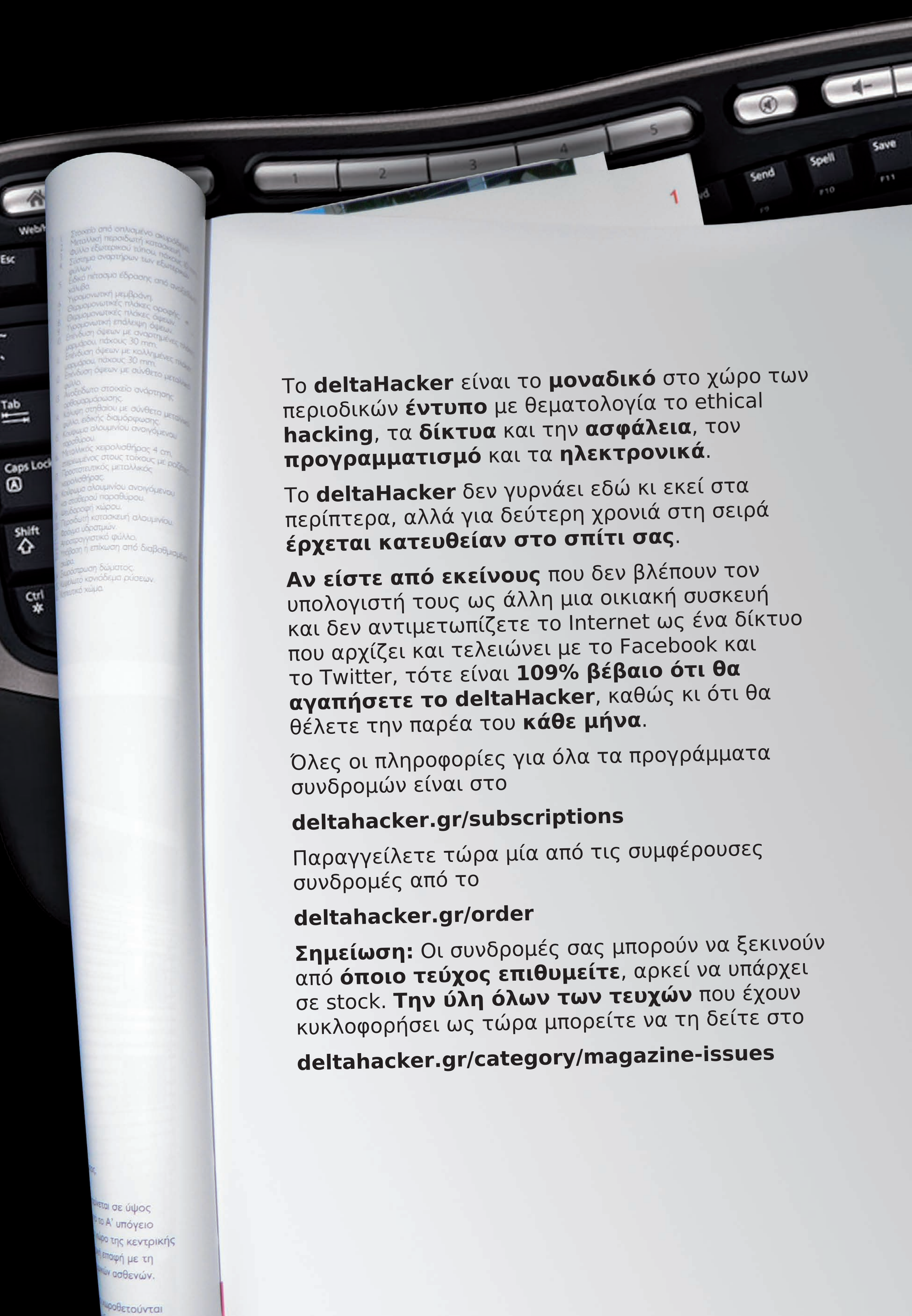



```
    potvalue = analogRead(A1);
    dutycycle = 64 + (potvalue >> 4);
    noisequality = 250 * (potvalue >> 4);
}
```

Ο κώδικας εδώ διαβάζει το άλλο ποτενσιόμετρο και ενημερώνει κατάλληλα τις dutycycle και noisequality. Οι δύο μεταβλητές ορίζονται με τέτοιο τρόπο, ώστε η dutycycle να βρίσκεται πάντα στην περιοχή [64...128], ενώ η noisequality στο [250...314]. Η περιοχή τιμών για τη δεύτερη μεταβλητή επιλέχθηκε σχεδόν αυθαίρετα. Το μοναδικό κριτήριο ήταν η χροιά του θορύβου που παράγεται για κάθε τιμή της noisequality. Για την άλλη μεταβλητή επιλέξαμε τη συγκεκριμένη περιοχή, αφού αυτή αντιστοιχεί σε duty cycle από 75% έως 50%. Για μεγαλύτερες τιμές του duty cycle, ο ήχος του τετραγωνικού κύματος αλλοιώνεται υπερβολικά. Όσο για τις μικρότερες τιμές του duty cycle, αυτές είναι απλά αδιάφορες! Σκεφτείτε ότι ένα τετραγωνικό κύμα με duty cycle στο 25%, αποτελεί ένα αντεστραμμένο τετραγωνικό κύμα με duty cycle στο 75%. Το αυτί μας όμως δεν μπορεί να αντιληφθεί αυτή την αντιστροφή. Έτσι, το άκουσμα των δύο κυμάτων δεν διαφέρει σε τίποτα. Αυτό ισχύει και για τετραγωνικά κύματα με duty cycle 45% και 55%, 40% και 60%, 35% και 65% κ.ο.κ.

Εικόνα 7
Το τσιπάκι ήχου των Commodore 64 (SID 6581) υπάρχει και σε shield! Οι φανατικοί των chiptunes μπορούν να γράφουν τη μουσική τους και στο Arduino ;)





Το **deltaHacker** είναι το **μοναδικό** στο χώρο των περιοδικών **έντυπο** με θεματολογία το ethical **hacking**, τα **δίκτυα** και την **ασφάλεια**, τον **προγραμματισμό** και τα **ηλεκτρονικά**.
Το **deltaHacker** δεν γυρνάει εδώ κι εκεί στα περίπτερα, αλλά για δεύτερη χρονιά στη σειρά **έρχεται κατευθείαν στο σπίτι σας**.
Αν είστε από εκείνους που δεν βλέπουν τον υπολογιστή τους ως άλλη μια οικιακή συσκευή και δεν αντιμετωπίζετε το Internet ως ένα δίκτυο που αρχίζει και τελειώνει με το Facebook και το Twitter, τότε είναι **109% βέβαιο ότι θα αγαπήσετε το deltaHacker**, καθώς κι ότι θα θέλετε την παρέα του **κάθε μήνα**.
Όλες οι πληροφορίες για όλα τα προγράμματα συνδρομών είναι στο
deltahacker.gr/subscriptions
Παραγγείλετε τώρα μία από τις συμφέρουσες συνδρομές από το
deltahacker.gr/order
Σημείωση: Οι συνδρομές σας μπορούν να ξεκινούν από **όποιο τεύχος επιθυμείτε**, αρκεί να υπάρχει σε stock. **Την ύλη όλων των τευχών** που έχουν κυκλοφορήσει ως τώρα μπορείτε να τη δείτε στο
deltahacker.gr/category/magazine-issues

Python & wxPython

THE GRAFFITI ARTIST!

Wrapping it up!

Στο προηγούμενο άρθρο ξεκινήσαμε τη δημιουργία μιας παιχνιδιάρικης εφαρμογής γραφικών και νομίζουμε ότι στην πορεία το διασκεδάσαμε αρκετά! Όχι μόνο γιατί η εφαρμογή που γράφουμε ταιριάζει απόλυτα στον χαρακτήρα μας, αλλά κυρίως γιατί το πραγματικό παιχνίδι —τουλάχιστον για εμάς— είναι η συγγραφή του κώδικα ;)



Skill: Beginner

Tags: Python, dictionaries, κλειδιά, τιμές, tuples, vector graphics

by Sonic

Πριν προχωρήσουμε στη βελτίωση του Graffiti Artist, αξίζει να κάνουμε μια μικρή ανακεφαλαίωση και να θυμηθούμε τα (ευχάριστα) προβλήματα που αντιμετωπίσαμε την προηγούμενη φορά.

- Απαλλαγήκαμε από τα σχήματα που σχεδιάζονται «μονοκονδυλιά»: Το νέο data model επιτρέπει να αποθηκεύουμε ένα σύνολο διαφορετικών γραμμών ως ένα ολοκληρωμένο σχήμα. Όταν ο χρήστης κάνει δεξί κλικ στον καμβά, η σχεδίαση ολοκληρώνεται και οι γραμμές που έχει σχεδιάσει ως εκείνη τη στιγμή αποθηκεύονται στο data model.
- Απαλλαγήκαμε από τη σχεδίαση στα τυφλά: Το πρόγραμμα εμφανίζει «preview» της γραμμής που πρόκειται να τραβήξουμε. Μάθαμε να χρησιμοποιούμε το περίεργο IDLE event!
- Προσθέσαμε ένα toolbar, καθώς και τη δυνατότητα επιλογής χρώματος. Φυσικά προσαρμόσαμε κατάλληλα και το data model, ώστε για κάθε γραμμή να αποθηκεύεται και το επιλεγμένο χρώμα.

Για τη συνέχεια έχουμε πολλές εργασίες ακόμα. Στο κάτω-κάτω, δεν θέλουμε μόνο να βελτιώσουμε το Graffiti Artist αλλά και να προσεγγίσουμε το μουσειακό Texas Artist! Ας δούμε τι άλλο πρέπει να γίνει.

- Θα πρέπει το πρόγραμμα να φτιάχνει ορθογώνια παραλληλόγραμμα και ελλείψεις (ή τετράγωνα και κύκλους, αν προτιμάτε). Καλό θα ήταν, εξάλλου, να υποστηρίζει και οποιοδήποτε άλλο σχήμα αποφασίσουμε να προσθέσουμε στο μέλλον.
- Θα πρέπει να αποθηκεύει και να φορτώνει το εκάστοτε έργο μας!
- Υπάρχουν μερικά κομμάτια κώδικα για το toolbar που έχουν γραφτεί πρόχειρα. Πρέπει να τα ξαναδούμε.

Όπως φαντάζεστε, ειδικά για την προσθήκη νέων σχημάτων το data model θα υποστεί ριζικές αλλαγές. Εμείς όμως θα ξεκινήσουμε από τα απλά. Πριν συνεχίσουμε, κατεβάστε τον κώδικα από το

http://bit.ly/dh019_gartist_final

Περισσότερες λειτουργίες - καλύτερος κώδικας

Το πιο εύκολο κομμάτι της εργασίας μας είναι να προσθέσουμε μερικά νέα κουμπάκια στο toolbar. Τα νέα εργαλεία που χρειαζόμαστε θα προστεθούν με λίγες μόνο γραμμές κώδικα. Θέλουμε ένα κουμπάκι για τις γραμμές, ένα για τις ελλείψεις, ένα για τα παραλληλόγραμμα και φυσικά ένα για το Load κι ένα για το Save[1]. Ας δούμε το σχετικό τμήμα του κώδικα:

```
self.toolbar.AddSeparator()
linetool = self.toolbar.AddTool(self.ID_LINE, wx.Bitmap("line.
png"), isToggle=True)
ellipsetool = self.toolbar.AddTool(self.ID_ELLIPSE,
wx.Bitmap("ellipse.png"), isToggle=True)
rectangletool = self.toolbar.AddTool(self.ID_RECTANGLE,
wx.Bitmap("rectangle.png"), isToggle=True)
self.toolbar.AddSeparator()
savetool = self.toolbar.AddTool(self.ID_SAVE, wx.Bitmap("save.png"))
```

1. Τα εικονίδια που επέλεξα για αυτές τις δύο λειτουργίες δείχνουν δισκέτες, οι οποίες δεν υπάρχουν πια. Μην γκρινιάζετε όμως! Πάλι καλά, να λέτε, που δεν έβαλα κασέτες ;)

```
loadtool = self.toolbar.AddTool(self.ID_LOAD, wx.Bitmap("load.
png"))
```

Για να δημιουργήσουμε τα νέα IDs που φαίνονται παραπάνω, χρησιμοποιήσαμε προηγουμένως την wx.NewID(). Το περισσότερο ενδιαφέρον όμως συγκεντρώνεται στις ακόλουθες γραμμές:

```
self.colordict = { self.ID_RED:(255,0,0),
                   self.ID_GREEN:(0,255,0),
                   self.ID_BLUE:(0,0,255),
                   self.ID_YELLOW:(255,255,0),
                   self.ID_PURPLE:(255,0,255),
                   self.ID_WHITE:(255,255,255),
                   self.ID_BLACK:(0,0,0) }
self.tooldict = { self.ID_LINE: 0,
                  self.ID_ELLIPSE: 1,
                  self.ID_RECTANGLE: 2 }
```

Αναρωτιέστε τι είναι όλα αυτά; Πρόκειται για μια δομή δεδομένων της Python που (περιέργως) δεν έχουμε χρησιμοποιήσει σε κανένα άλλο πρόγραμμα. Πρόκειται για ένα dictionary — ή λεξικό, αν προτιμάτε. Φανταστείτε ένα κανονικό λεξικό (π.χ., ένα Άγγλο-Ελληνικό) και θα καταλάβετε πολύ εύκολα τη λειτουργία τους. Αυτό που κάνει ένα λεξικό είναι να αντιστοιχεί λέξεις της μίας γλώσσας σε λέξεις της άλλης. Τα Python dictionaries κάνουν το ίδιο πράγμα: Αντιστοιχούν ένα κλειδί σε μια τιμή. Κάθε ζεύγος κλειδιού-τιμής χωρίζεται από τον χαρακτήρα «:», ενώ όλες οι αντιστοιχίες που περιλαμβάνει το λεξικό δηλώνονται μέσα σε άγκιστρα[2]. Έχοντας δηλώσει τα παραπάνω λεξικά, αν γράψουμε κάτι τέτοιο:

```
print self.colordict[self.ID_RED]
```

…θα πάρουμε την ακόλουθη απάντηση:

```
(255,0,0)
```

Βλέπετε, σύμφωνα με το λεξικό μας, το κλειδί self.ID_RED αντιστοιχεί στο tuple (255,0,0). Δείτε τώρα πόσο πιο όμορφα μπορούμε να γράψουμε τη συνάρτηση OnSetColor, απομακρύνοντας πλέον τα απανωτά if:

```
def OnSetColor(self, e):
  commandId = e.GetId()
  for toolId in range(self.ID_RED, self.ID_BLACK+1):
    self.toolbar.ToggleTool(toolId, 0)
  self.toolbar.ToggleTool(commandId, 1)
  self.drawpanel.currentcolor=wx.Colour(self.colordict[commandId][0],
                                        self.colordict[commandId][1],
                                        self.colordict[commandId][2])
```

Αν το commandId έχει την τιμή self.ID_RED, το self.colordict[commandId] θα επιστρέψει το tuple (255,0,0). Ωστόσο ο constructor wx.Colour δεν δέχεται tuples στην είσοδό του, οπότε χωρίζουμε τις τιμές σε Red, Green και Blue. Έτσι, το self.colordict[commandId][0], για παράδειγμα, θα μας

2. Θυμηθείτε ότι οι αγκύλες χρησιμοποιούνται στις λίστες και οι παρενθέσεις στα tuples.

δώσει το 255. Με την ίδια λογική αλλάζουμε και τη συνάρτηση για τα νέα εργαλεία (γραμμή, έλλειψη και παραλληλόγραμμο):

```
def OnSetTool(self, e):
  commandId = e.GetId()
  for toolId in range(self.ID_LINE, self.ID_RECTANGLE+1):
    self.toolbar.ToggleTool(toolId,0)
  self.toolbar.ToggleTool(commandId, 1)
  self.drawpanel.thetool = self.tooldict[commandId]
```

Στην κλάση DrawPanel έχουμε ορίσει ένα νέο member variable, το thetool. Η τιμή του καθορίζει το εργαλείο σχεδίασης που έχουμε επιλέξει. Κάπως αυθαίρετα, έχουμε κάνει την παρακάτω αντιστοίχιση:

Εργαλείο Σχεδίασης	Τιμή thetool
0	Γραμμή
1	Έλλειψη
2	Παραλληλόγραμμο

Αν αναρωτιέστε πότε και που ορίσαμε αυτήν την αντιστοιχία, δείτε το λεξικό ονόματι self.tooldict, που ορίσαμε λίγο παραπάνω! Αυτό που δεν έχουμε ολοκληρώσει ακόμα είναι οι

λειτουργίες Save και Load. Όμως, πριν φτάσουμε να τις υλοποιήσουμε, πρέπει να δούμε πολλά πράγματα ακόμα. Αυτές οι δύο συναρτήσεις βασίζονται στο data model, το οποίο θα χρειαστεί να αλλάξει αρκετά για να καλύψει τις νέες μας ανάγκες.

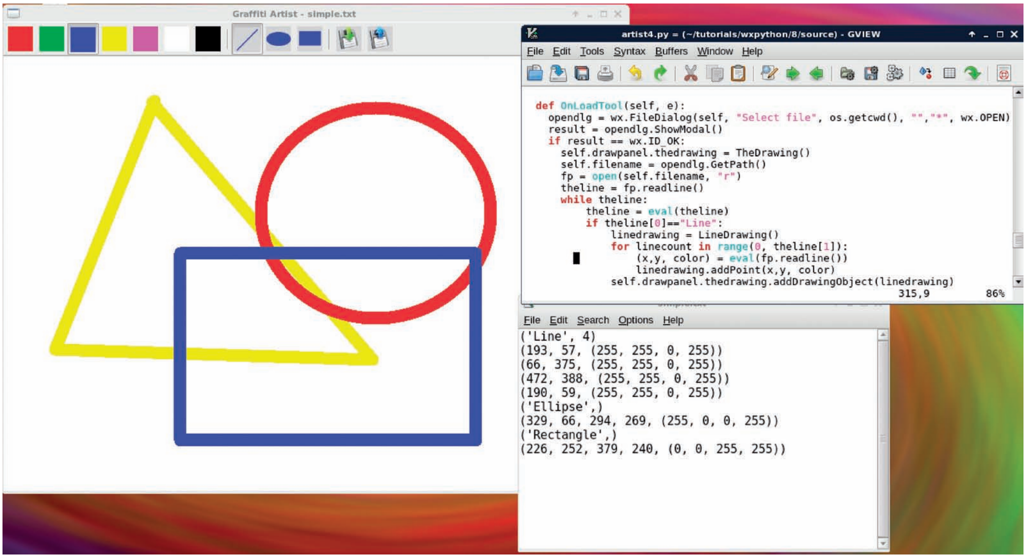
Data Model revisited!

Πριν δούμε ποιες αλλαγές χρειαζόμαστε στο data model, καλό είναι να ρίξουμε μια ματιά στην τρέχουσα μορφή του. Καθώς μια εικόνα αξίζει 1024 λέξεις, δείτε το σχήμα 1. Στο τρέχον data model ισχύουν τα ακόλουθα.

- Έχουμε ένα αντικείμενο τύπου PointList, το self.pointlist. Αυτό περιέχει μια λίστα, η οποία με τη σειρά της περιέχει tuples. Κάθε tuple έχει τρεις τιμές: Δυο συντεταγμένες κι ένα χρώμα. Κάθε φορά που τραβάμε μια γραμμή, η λίστα του αντικειμένου αυξάνεται με την προσθήκη του αντίστοιχου tuple. Όταν κάνουμε δεξί κλικ δηλώνουμε ότι έχουμε ολοκληρώσει το σχήμα και προστίθεται το τελευταίο σημείο στη λίστα.
- Με την ολοκλήρωση ενός σχήματος αποθηκεύουμε μια αναφορά για αυτό το αντικείμενο στη λίστα του self.linedrawing.
- Επιπρόσθετα, με την ολοκλήρωση ενός σχεδίου δημιουργούμε ένα νέο αντικείμενο τύπου PointList, στο οποίο δείχνει πλέον το self.pointlist. Έτσι, προετοιμαζόμαστε για τη δημιουργία του επόμενου σχήματος.

Ενδιαφέρον είναι το γεγονός ότι για να σχεδιάσουμε εκ νέου την περιοχή εργασίας (στη συνάρτηση OnPaint), πρέπει να κάνουμε δύο κινήσεις:

- Να διαβάσουμε όλα τα αντικείμενα τύπου PointList που έχουμε



Το Graffiti Artist εν δράσει: Το σχέδιο που βλέπετε περιέχεται στο αρχείο simple.txt, το οποίο μπορείτε να ανοίξετε είτε στο graffiti artist είτε σε κάποιον text editor. Ένα πλεονέκτημα των διανυσματικών γραφικών είναι ότι μπορούν να αναπαρασταθούν εύκολα με τη βοήθεια λίγων αριθμών.

αποθηκεύσει στη λίστα του LineDrawing και να τραβήξουμε τις αντίστοιχες γραμμές.

- Να δούμε αν υπάρχει κάποιο "ξεκρέμαστο" PointList —κάποιο μισοτελειωμένο σχήμα— και να σχεδιάσουμε κι αυτές τις γραμμές. Σε αντίθεση με το παραλληλόγραμμο και την έλλειψη που θα προσθέσουμε σε λίγο, το εργαλείο γραμμών είναι αρκετά πιο πολύπλοκο, καθώς ο επίδοξος καλλιτέχνης μπορεί να τραβήξει, διαδοχικά, όσες γραμμές θέλει!

Ας υποθέσουμε τώρα ότι το πρόγραμμα πρέπει να χειρίζεται παραλληλόγραμμα και ελλείψεις. Από σχεδιαστικής άποψης, αυτά τα δύο αντικείμενα είναι αρκετά απλά και περιγράφονται από τα ακόλουθα στοιχεία.

- Ένα ζεύγος συντεταγμένων
- Πλάτος και ύψος
- Χρώμα

Προφανώς, αυτά τα στοιχεία θα πρέπει επίσης να αποθηκεύονται στο data model. Οι σχετικές αλλαγές φαίνονται στο σχήμα 2.

- Στη μέχρι τώρα μορφή του μοντέλου, ο κύριος αποθηκευτικός χώρος ήταν το αντικείμενο LineDrawing: Σε αυτό αποθηκεύονταν παραπομπές για όλα τα σχέδια.
- Στο νέο μοντέλο, δεν υπάρχει μόνο ένα είδος αντικειμένου. Επομένως, ο ρόλος που είχε το αντικείμενο τύπου LineDrawing θα μεταφερθεί τώρα στο αντικείμενο TheDrawing (πρωτότυπο όνομα, δεν μπορείτε να πείτε).
- Ο κώδικας που έχουμε γράψει μέχρι στιγμής για το χειρισμό του αντικειμένου LineDrawing, αντιστοιχεί στο πρώτο εργαλείο του toolbar. Όταν ολοκληρώνεται ένα σχήμα που αποτελείται από γραμμές, η αναφορά self.linedrawing αποθηκεύεται στο αντικείμενο τύπου TheDrawing ενώ το ίδιο το self.linedrawing δείχνει σ'

ένα κενό αντικείμενο LineDrawing και είναι έτοιμο να δεχτεί το επόμενο σχήμα.

- Το αντικείμενο τύπου TheDrawing, εκτός από αναφορές σε αντικείμενα του ιδίου τύπου, μπορεί να περιέχει και αναφορές σε αντικείμενα τύπου RectangleDrawing και EllipseDrawing.

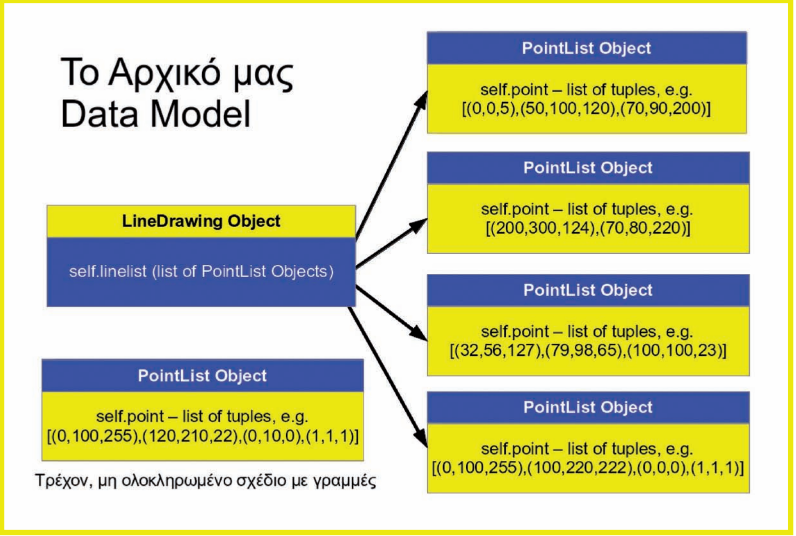
Μετά από αυτήν την αλλαγή, για να λειτουργήσει σωστά η OnPaint και να σχεδιάσει εκ νέου την οθόνη και τα περιεχόμενά της, πρέπει να γίνονται τα εξής:

- Να διατρέχουμε τη λίστα αντικειμένων που περιέχεται στο αντικείμενο τύπου TheDrawing.
- Να ελέγχουμε αν κάθε στοιχείο που διαβάζουμε είναι τύπου LineDrawing, RectangleDrawing ή EllipseDrawing.
- Να διαβάζουμε τα δεδομένα του συγκεκριμένου αντικειμένου: Τη λίστα με τα σημεία και τα χρώματα, αν πρόκειται για LineDrawing, ή τις συντεταγμένες, το μέγεθος και το χρώμα, αν πρόκειται για παραλληλόγραμμο ή έλλειψη.
- Με βάση τα δεδομένα, να σχεδιάζουμε το σωστό σχήμα.
- Φυσικά, αν υπάρχει κάποιο "ξεκρέμαστο" LineDrawing (ένα μισοτελειωμένο σχήμα γραμμών) θα πρέπει επίσης να σχεδιαστεί. Τα άλλα δύο σχήματα (παραλληλόγραμμο και έλλειψη) φτιάχνονται σε δύο μόνο κινήσεις και δεν υπάρχει κίνδυνος να μείνουν μισοτελειωμένα.

Έχοντας ξεκαθαρίσει στη σκέψη μας τα παραπάνω, είμαστε σχεδόν έτοιμοι να γράψουμε την OnPaint στην ανανεωμένη της μορφή.

```
for drawing in self.thedrawing.drawingobjects:
  if isinstance(drawing, LineDrawing):
    (x,y,color) = drawing.point[0]
    dc.SetPen(wx.Pen(color,16))
    for thepoint in drawing.point:
      dc.SetPen(wx.Pen(thepoint[2],16))
      dc.DrawLine(x,y, thepoint[0], thepoint[1])
      x = thepoint[0]
      y = thepoint[1]
  if isinstance(drawing, RectangleDrawing):
    (x,y, width, height, color) = drawing.getRectangle()
    dc.SetPen(wx.Pen(color,16))
    dc.DrawRectangle(x,y,width,height)

  if isinstance(drawing, EllipseDrawing):
    (x,y, width, height, color) = drawing.getEllipse()
    dc.SetPen(wx.Pen(color,16))
    dc.DrawEllipse(x,y,width,height)
```



Το αρχικό data model, τότε που το graffiti artist σχεδίαζε μόνο γραμμές. Εδώ, κάθε αντικείμενο τύπου PointList αποθηκεύει ένα πλήρες σχήμα με γραμμές. Το δε LineDrawing διατηρεί μια λίστα με αναφορές σε αντικείμενα PointList. Η σχεδίαση της οθόνης απαιτεί να επεξεργαστούμε τη λίστα που περιέχει το LineDrawing κι ενδεχομένως το PointList (για σχήματα που δεν έχουν ολοκληρωθεί ακόμα).

Το αντικείμενο self.thedrawing ανήκει στην κλάση TheDrawing και, όπως είπαμε, αποτελεί πλέον τη βασική "αποθήκη" για τα σχήματα. Περιέχει μια λίστα που ονομάζεται drawingobjects και στην οποία αποθηκεύονται αναφορές των τριών άλλων αντικειμένων σχεδίασης. Αναρωτιέστε πώς ξεχωρίζουμε μεταξύ τους τις διάφορες αναφορές; Μα, είναι πολύ απλό. Δείτε:

```
for drawing in self.thedrawing.drawingobjects:
  if isinstance(drawing, LineDrawing):
```

Διατρέχουμε τη λίστα των αποθηκευμένων αντικειμένων και χρησιμοποιούμε τη συνάρτηση isinstance, για να τσεκάρουμε τον τύπο του εκάστοτε αντικειμένου! Για να δουλέψει αυτό το κόλπο πρέπει οι κλάσεις μας να προέρχονται από τη γονική κλάση object, μια αλλαγή που έχουμε κάνει ήδη αλλά δεν φαίνεται στο παραπάνω απόσπασμα :) Κατά τα άλλα, ο κώδικας δεν διαφέρει ιδιαίτερα από την προηγούμενη εκδοχή της OnPaint.

Ελλείψεις και Παραλληλόγραμμα

Τι περιέχει μια κλάση για ελλείψεις; Μη φανταστείτε κάτι σπουδαίο. Αν εξαιρέσουμε το όνομα, είναι ακριβώς ίδια με εκείνη για τα παραλληλόγραμμα:

```
class EllipseDrawing(object):
  def __init__(self, x, y, width, height, color):
    self.x = x
    self.y = y
    self.width = width
    self.height = height
    self.color = color
  def getEllipse(self):
    return (self.x, self.y, self.width, self.height, self.color)
```

Στον constructor δίνουμε μόνο τις συντεταγμένες του επάνω αριστερού σημείου, το πλάτος και το ύψος του σχήματος, όπως επίσης και το χρώμα του. Έχουμε και μια συνάρτηση getEllipse, που αναλαμβάνει να επιστρέψει όλα τα παραπάνω δεδομένα στη βολική μορφή ενός tuple. Όλες οι συναρτήσεις που σχετίζονται με τη σχεδίαση πρέπει τώρα να προσαρμοστούν ανάλογα. Αυτό όμως είναι εύκολο γιατί μπορούμε να γνωρίζουμε ποιο εργαλείο χρησιμοποιείται, ελέγχοντας κάθε φορά τη μεταβλητή thetool. Ο κώδικας που γράψαμε στο προηγούμενο άρθρο αντιστοιχεί στο εργαλείο γραμμής (thetool = 0). Γι' αυτό το λόγο, οι συναρτήσεις όπως η LeftClick ξεκινούν πλέον με τις ακόλουθες γραμμές:

```
def LeftClick(self, e):
  dc = wx.ClientDC(self)
  x = e.GetX()
  y = e.GetY()
  if self.thetool == 0:
    self.MyLine(dc,x,y)
    self.linedrawing.addPoint(x,y, self.currentcolor)
...
```

Για το παραλληλόγραμμο και την έλλειψη, ο κώδικας είναι εξίσου απλός:

```
if self.thetool == 1 and self.startx == -1:
  dc.SetPen(wx.Pen(self.currentcolor,16))
  dc.SetBrush(wx.Brush("black", wx.TRANSPARENT))
  dc.DrawEllipse(x,y,1,1)
  self.startx = x
  self.starty = y
```

Έχουμε ορίσει δύο μεταβλητές (self.startx και self.starty) στις οποίες αποθηκεύουμε τις συντεταγμένες όπου έγινε το αριστερό κλικ, για να ξεκινήσει το παραλληλόγραμμο ή η έλλειψη. Η αρχική τιμή τους είναι -1 και δείχνει ότι ο χρήστης δεν έχει αρχίσει να σχεδιάζει ακόμα. Αν όμως η τιμή είναι διαφορετική, το αριστερό κλικ δεν κάνει τίποτα: Πρέπει πρώτα να ολοκληρώσουμε τη σχεδίαση με ένα δεξί κλικ. Χρησιμοποιούμε την OnMotion και την OnIdle με παρόμοιο τρόπο, για να εμφανίσουμε ένα "preview" του σχήματος. Ας δούμε λίγο τη συνάρτηση RightClick:

```
if self.thetool == 1 and self.startx != -1:
  dc = wx.ClientDC(self)
  dc.SetPen(wx.Pen(self.currentcolor,16))
  dc.SetBrush(wx.Brush("black",wx.TRANSPARENT))
```

Για να βρούμε το πλάτος και το ύψος (που απαιτούνται από τις συναρτήσεις dc.DrawEllipse και dc.DrawRectangle) έχουμε γράψει μια πολύ απλή συνάρτηση (την getWidthHeight) που δέχεται αρχικές και τελικές συντεταγμένες κι επιστρέφει ένα tuple με το πλάτος και το ύψος του σχήματος:

```
(width, height) = getWidthHeight(self.startx, self.starty, x, y)
dc.DrawEllipse(self.startx, self.starty, width, height)
```

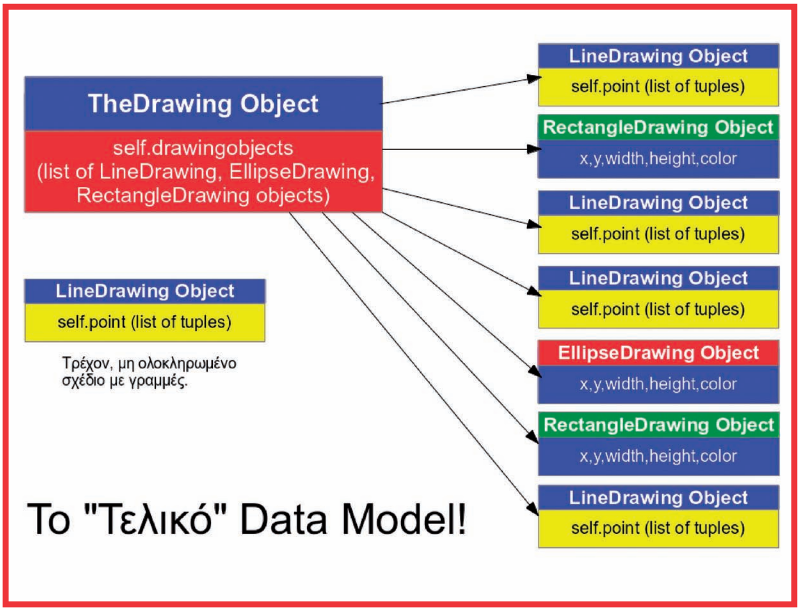
Αμέσως μόλις σχεδιαστεί η έλλειψη δημιουργούμε ένα αντικείμενο τύπου EllipseDrawing και αποθηκεύουμε την αναφορά του στο self.thedrawing:

```
    theellipse = EllipseDrawing(self.startx, self.starty, width,
  height, self.currentcolor)
    self.thedrawing.addDrawingObject(theellipse)
```

Τέλος, θέτουμε το startx=-1. Έτσι σηματοδοτούμε το τέλος της σχεδίασης της έλλειψης και, ταυτόχρονα, το πρόγραμμα ξέρει ότι δεν έχει ξεκινήσει η σχεδίαση μιας καινούργιας.

```
    self.startx = -1
```

Ο κώδικας για το παραλληλόγραμμο είναι ...συναρπαστικά όμοιος.



Στην εξελιγμένη του μορφή, το data model μπορεί να χειριστεί γραμμές, ελλείψεις και παραλληλόγραμμα, ενώ επεκτείνεται και εύκολα. Το βασικό αντικείμενο στο οποίο αποθηκεύονται τα σχήματα ανήκει πλέον στην κλάση TheDrawing. Το ρόλο της κλάσης PointList έχει πάρει η κλάση LineDrawing, ενώ έχουν δημιουργηθεί και οι αντίστοιχες EllipseDrawing και RectangleDrawing. Διαβάζοντας τον κώδικα θα διαπιστώσετε ότι η συνάρτηση isinstance είναι πολύ χρήσιμη: Επιτρέπει να δούμε τι είδους αντικείμενο έχουμε αποθηκεύσει μέσα στο TheDrawing!

Save και Load: Τελικά είναι απλοί αριθμοί!

Πιστεύουμε να έχετε καταλάβει ότι το πρόγραμμα δημιουργεί διανυσματικά γραφικά (vector graphics). Κάθε σχέδιο περιγράφεται από μια σειρά αριθμών: Συντεταγμένες, χρώματα, μεγέθη και είδος αντικείμενου. Όλα αυτά είναι αποθηκευμένα όμορφα και τακτοποιημένα μέσα στο data model του προγράμματος. Επομένως, για τη λειτουργία της αποθήκευσης αρκεί να φροντίσουμε για την εγγραφή τους σε ένα αρχείο. Αντίστοιχα, για τη διαδικασία της φόρτωσης θα πρέπει να ενημερώσουμε κατάλληλα το data model και να δημιουργήσουμε τα αντίστοιχα αντικείμενα, διαβάζοντας τις τιμές από το αρχείο. Το ωραίο είναι ότι μόλις πετύχουμε αυτή τη μεταφορά, η OnPaint θα αναλάβει να σχεδιάσει τα πάντα στην οθόνη και δεν θα χρειαστεί κάτι άλλο από μέρους μας! Ένας απλός τρόπος αποθήκευσης φαίνεται παρακάτω:

```
fp = open(self.filename, "w")
```

Με αυτή τη γραμμή ανοίγουμε ένα αρχείο για εγγραφή. Φυσικά, το self.filename το επιλέγουμε με τη βοήθεια ενός common dialog — διαδικασία που είδαμε στον wxEditor (βλ. deltaHacker από 013 έως 015). Διατρέχουμε έπειτα τη λίστα αντικειμένων και γράφουμε τα δεδομένα σε μορφή tuples:

```
for drawing in self.drawpanel.thedrawing.drawingobjects:
  if isinstance(drawing, LineDrawing):
    fp.write("('Line', %s)\n" % len(drawing.point))
    for point in drawing.point:
      fp.write("(%s, %s, %s) \n" % (point[0], point[1],
point[2]))
  if isinstance(drawing, RectangleDrawing):
    (x,y, width, height, color) = drawing.getRectangle()
    fp.write("('Rectangle',)\n")
    fp.write("(%s, %s, %s, %s, %s)\n" % (x,y,width,height,color))
  if isinstance(drawing, EllipseDrawing):
    (x,y, width, height, color) = drawing.getEllipse()
    fp.write("('Ellipse',)\n")
    fp.write("(%s, %s, %s, %s, %s)\n" % (x,y,width,height,color))
fp.close()
```

Για παράδειγμα, δείτε τι θα αποθηκεύσει το παραπάνω απόσπασμα κώδικα, για ένα κίτρινο τρίγωνο (τρεις γραμμές), έναν κόκκινο κύκλο κι ένα μπλε παραλληλόγραμμο3:

```
 ('Line', 4)
(193, 57, (255, 255, 0, 255))
(66, 375, (255, 255, 0, 255))
(472, 388, (255, 255, 0, 255))
(190, 59, (255, 255, 0, 255))
('Ellipse',)
(329, 66, 294, 269, (255, 0, 0, 255))
('Rectangle',)
(226, 252, 379, 240, (0, 0, 255, 255))
```

Ας δούμε το πρώτο tuple:

```
('Line',4)
```

Όπως καταλαβαίνετε, δηλώνει ότι οι επόμενες 4 γραμμές περιέχουν tuples με τις συντεταγμένες και το χρώμα των γραμμών που απαρτίζουν ένα σχήμα. Οι ελλείψεις και τα παραλληλόγραμμα περιέχουν μια μόνο εντολή σχεδίασης. Έτσι, στην περίπτωση αυτών των σχημάτων, δεν χρειάζεται να γράφουμε το πλήθος των γραμμών που ακολουθούν. Το σχετικό tuple είναι πολύ απλό και δηλώνει μόνο τον τύπο του σχήματος:

```
('Ellipse',)
```

Τα δεδομένα ενός τέτοιου σχήματος ακολουθούν σε ένα σαφώς μεγαλύτερο tuple:

```
(329, 66, 294, 269, (255, 0, 0, 255))
```

3. Τα περιεχόμενα που φαίνονται ανήκουν στο αρχείο simple.txt και θα το βρείτε στο πακέτο με τον κώδικα του άρθρου. Αυτό το αρχείο μπορείτε να το φορτώσετε και στην τελική έκδοση του Graffiti Artist.

Αναρωτιέστε γιατί μπήκαμε στον κόπο να γράψουμε τα δεδομένα με τη μορφή tuples; Για να πετύχουμε κάτι τέτοιο, χρειάστηκε να προσθέσουμε κόμματα και παρενθέσεις! Όπως θα δείτε σύντομα, με αυτόν τον τρόπο αποθήκευσης υπάρχουν σαφή πλεονεκτήματα κατά τη διαδικασία φόρτωσης. Ας δούμε το σχετικό απόσπασμα κώδικα:

```
fp = open(self.filename, "r")
theline = fp.readline()
while theline:
    theline = eval(theline)
    if theline[0]=="Line":
        linedrawing = LineDrawing()
        for linecount in range(0, theline[1]):
            (x,y, color) = eval(fp.readline())
            linedrawing.addPoint(x,y, color)
        self.drawpanel.thedrawing.addDrawingObject(linedrawing)

    if theline[0]=="Rectangle":
        (x,y, width, height, color)= eval(fp.readline())
        therectangle = RectangleDrawing(x,y,width,height,color)
        self.drawpanel.thedrawing.addDrawingObject(therectangle)

    if theline[0]=="Ellipse":
        (x,y,width, height, color) = eval(fp.readline())
        theellipse = EllipseDrawing(x,y,width,height,color)
        self.drawpanel.thedrawing.addDrawingObject(theellipse)

    theline = fp.readline()

fp.close()
```

Η συνάρτηση eval της Python επιτρέπει να μετατρέψουμε απευθείας σε tuples, τις γραμμές που διαβάζουμε από το αρχείο μέσω της fp.readline(). Έτσι, αντιστρέφουμε τη διαδικασία του Save πολύ εύκολα: Διαβάζουμε το αρχείο γραμμή προς γραμμή, παίρνουμε τα αντίστοιχα tuples, δημιουργούμε τα κατάλληλα αντικείμενα και αποθηκεύουμε τις αναφορές τους στο αντικείμενο thedrawing. Ας δούμε ένα παράδειγμα: Διαβάζοντας την πρώτη γραμμή, η eval θα μας δώσει ένα tuple:

```
theline[0] = "Line"
theline[1] = 4
```

Τώρα μπορούμε να δημιουργήσουμε ένα αντικείμενο linedrawing και να διαβάζουμε τις επόμενες 4 γραμμές του αρχείου, οι οποίες γνωρίζουμε ότι περιέχουν πληροφορίες για γραμμές (tuples με συντεταγμένες και χρώμα):

```
if theline[0]=="Line":
  linedrawing = LineDrawing()
  for linecount in range(0, theline[1]):
    (x,y, color) = eval(fp.readline())
```

Στη συνέχεια, κάθε tuple που διαβάζουμε από τις ακόλουθες γραμμές του αρχείου, αποθηκεύεται στο linedrawing:

```
      linedrawing.addPoint(x,y, color)
```

Τέλος, προσθέτουμε μια αναφορά για το ολοκληρωμένο αντικείμενο linedrawing, στη λίστα του αντικειμένου thedrawing:

```
    self.drawpanel.thedrawing.addDrawingObject(linedrawing)
```

Η διαδικασία για την έλλειψη και το παραλληλόγραμμο είναι ακόμα πιο απλή. Πιστεύουμε ότι δεν θα δυσκολευτείτε να την κατανοήσετε.

Επίλογος

Τελειώνοντας το Graffiti Artist, ολοκληρώνουμε κι αυτή τη σειρά άρθρων για την Python και την wxPython. Μη φανταστείτε όμως ότι το μέλλον δεν περιλαμβάνει περισσότερο programming στην αγαπημένη μας γλώσσα! Στο μεταξύ, σας καλούμε να τροποποιήσετε τα προγράμματα που δημιουργήσαμε μέσα από τη σειρά των άρθρων. Αν θέλετε μερικές ιδέες για βελτιώσεις του Graffiti Artist, δοκιμάστε κάτι από τα ακόλουθα.

- Προσθέστε εργαλείο για ρύθμιση πάχους γραμμών
- Προσθέστε εργαλεία για δημιουργία ελλείψεων και παραλληλογράμμων, γεμισμένων με χρώμα
- Προσθέστε εργαλεία για περισσότερα έτοιμα σχήματα. Θα μπορούσατε να ξεκινήσετε με κύκλους και τετράγωνα, ειδικές περιπτώσεις των σχημάτων που έχουμε ήδη.
- Δώστε στο πρόγραμμα τη δυνατότητα αποθήκευσης των σχεδίων σε αρχεία κάποιου τύπου εικόνας (π.χ., σε JPEG ή σε PNG).
- Βελτιώστε το κομμάτι Save/Load. Για παράδειγμα, στην τρέχουσα μορφή του προγράμματος δεν γίνεται πουθενά έλεγχος για την εγκυρότητα του αρχείου.
- Μπορείτε να προσθέσετε έξτρα λειτουργίες με μενού, διαλόγους επιλογών και άλλα καλούδια που θα φτιάξετε στο wxGlade.

Μετά από όλα αυτά, μένει να σας ευχηθούμε happy pythoning και πολλά δημιουργικά απογεύματα!


DeltaHacker
Admin Panel
ΑΝάλυση Πακέτων
8,307
Είμαστε κοινωνικοί — ειδικά στον online κόσμο.
Είμαστε αστείοι, ευφυείς και φινετσάτοι — ειδικά στον online κόσμο.
Λέγεται ότι έχουμε καβαλήσει το καλάμι — τόσο στον online, όσο και στον offline κόσμο.
Όπως και να 'χει, το διαχειριζόμαστε γενναία.
Περάστε από τα στέκια μας να τα πούμε — ή να μας την πείτε.
twitter.com/deltaHacker
facebook.com/deltaHacker
HACKER



Skill: Intermediate

Tags: Arduino, Nunchuk, I2C, libraries, wire

Σπιτικές βιβλιοθήκες για το Arduino!

by GiannouG

Υπάρχουν εκατοντάδες —αν όχι χιλιάδες— βιβλιοθήκες για το Arduino. Πολλές από αυτές όμως είναι κακογραμμένες ή απλά στερούνται ορισμένων δυνατοτήτων που εμείς θέλουμε. Στο κάτω-κάτω, από τη στιγμή που δεν τις έχουμε φτιάξει με τα χεράκια μας, αποκλείεται να είναι άξιες για να ενσωματωθούν στα δικά μας προγράμματα! Μήπως λοιπόν πρέπει να αρχίσουμε να κατασκευάζουμε τις δικές μας;

Τα τελευταία χρόνια το Arduino έχει κυριαρχήσει ως το πιο εύχρηστο development board ηλεκτρονικών. Σε αυτό έχουν συμβάλλει τόσο ο open source χαρακτήρας του, όσο και η πληθώρα βιβλιοθηκών που του επιτρέπουν να επικοινωνεί με εκατοντάδες περιφερειακά. Για κάθε ηλεκτρονική συσκευή που θα μπορούσε να συνδεθεί σε μικροελεγκτή, υπάρχει σίγουρα μια βιβλιοθήκη που επιτρέπει τον άμεσο και εύκολο χειρισμό από το Arduino. Βέβαια, οι βιβλιοθήκες δημιουργούνται συνήθως από προγραμματιστές της κοινότητας, οι οποίοι δεν έχουν απαραίτητα περισσότερες γνώσεις από εμάς. Επιπρόσθετα, όσο πιο δημοφιλές είναι κάποιο περιφερειακό, τόσο περισσότερες είναι και οι αντίστοιχες βιβλιοθήκες. Μέσα σε αυτό το χαμό είναι πολύ εύκολο να πέσουμε σε μια προβληματική βιβλιοθήκη, που είτε γράφτηκε γρήγορα και απρόσεκτα είτε γράφτηκε για μια παλιότερη έκδοση του αντίστοιχου hardware. Εξάλλου, ορισμένες φορές θα χρειαστούμε μια λειτουργία, η οποία απουσιάζει από τις υπάρχουσες βιβλιοθήκες. Όπως καταλαβαίνετε, όλα αυτά είναι λογικό να μας βάζουν σε σκέψεις...

Μήπως κάθε φορά που χρειαζόμαστε μια βιβλιοθήκη, πρέπει να περιοριζόμαστε στις έτοιμες λύσεις και απλά να ψάχνουμε τη λιγότερο κακή; Μήπως πάλι να γράφαμε πάντα τις δικές μας βιβλιοθήκες και να ξεμπερδεύαμε; Αλήθεια, πώς μπορούμε να δημιουργήσουμε μια βιβλιοθήκη από το μηδέν; Σε τι γλώσσα μπορούμε να γράψουμε τον κώδικα; Ποια είναι η *προβλεπόμενη* δομή για μια βιβλιοθήκη; Νομίζουμε ότι αυτές οι απορίες έχουν περάσει από το μυαλό όλων όσοι ασχολούνται αρκετά με το Arduino. Για να 'μαστε ειλικρινείς, ορισμένες από τις ερωτήσεις μπορούν να δεχτούν πολλές διαφορετικές απαντήσεις, ανάλογα με την οπτική του καθενός. Εμείς όμως το 'χουμε πάρει απόφαση: Κατά κανόνα, οι βιβλιοθήκες που κυκλοφορούν δεν μας καλύπτουν. Εμείς θέλουμε βιβλιοθήκες που όχι μόνο θα λειτουργούν άψογα, αλλά θα ακολουθούν και τα πρότυπα. Αυτό δεν αποτελεί ιδιοτροπία! Προτιμάμε τις βιβλιοθήκες που γράφτηκαν σύμφωνα με τα πρότυπα γιατί είναι εύκολα επεκτάσιμες, τόσο από εμάς τους ίδιους όσο και από άλλους. Εμείς, βλέπετε, αγαπάμε και σεβόμαστε πραγματικά το Ανοιχτό Λογισμικό[1].

Εν αρχή ην το hardware

Η συντριπτική πλειονότητα των βιβλιοθηκών του Arduino εξυπηρετούν τον εύκολο χειρισμό κάποιας συσκευής. Είναι ελάχιστες εκείνες που αντιμετωπίζουν θεωρητικά ζητήματα και περιορίζονται στην υλοποίηση κάποιων αλγορίθμων. Ακριβώς γι' αυτό, η γνωριμία μας με το θέμα θα ξεκινήσει δημιουργώντας μια βιβλιοθήκη για τον εύκολο χειρισμό μιας συσκευής. Μάλιστα, για να κάνουμε το θέμα ακόμα πιο ενδιαφέρον, θα ασχοληθούμε με τη διασύνδεση του Wii Nunchuk. Πρόκειται για ένα πρόσθετο του WiiMote, της κονσόλας Nintendo Wii. Το Nunchuk συνδέεται ενσύρματα στο χειριστήριο του Wii (WiiMote) και προσθέτει επιπλέον χειριστήρια. Το Nunchuk επικοινωνεί με το πρωτόκολλο I2C και το Arduino (συγκεκριμένα ο μικροελεγκτής AVR) διαθέτει εξειδικευμένο hardware ειδικά για αυτό το είδος επικοινωνίας. Επιπλέον, το περιβάλλον του Arduino περιλαμβάνει την βιβλιοθήκη Wire, η οποία διευκολύνει σημαντικά την επικοινωνία με συσκευές I2C!

Κάπως έτσι ξεκινάει η περιπέτειά μας. Πρώτα απ' όλα πρέπει να μάθουμε όσο πιο πολλά μπορούμε για το χειριστήριο. Μπορεί να γνωρίζουμε πώς επικοινωνεί, αλλά δεν έχουμε ιδέα για το πώς ακριβώς συνδέεται

1. Παρ' όλο που προτιμάμε το Ubuntu κι όχι το Fedora ;)



με το Arduino, ούτε πώς μπορούμε να αντλήσουμε δεδομένα απ' αυτό. Όπως φαντάζεστε, επειδή πρόκειται για ένα περιφερειακό κονσόλας είναι εξαιρετικά δύσκολο να βρούμε datasheet[2]. Έτσι, θα βασιστούμε στο Google και στην έρευνα που έχουν κάνει άλλοι. Επιπλέον, θα χρειαστεί να ρίξουμε μια ματιά στη βιβλιοθήκη Wire, για να δούμε ποιες διαδικασίες της θα χρειαστούμε.

Το I2C χρησιμοποιεί δύο καλώδια για την επικοινωνία και συνολικά τέσσερα, αν συνυπολογίσουμε την τροφοδοσία της συσκευής. Επομένως, το πρώτο που πρέπει να κάνουμε είναι να κόψουμε το βύσμα του χειριστηρίου! Τα καλώδια του διαύλου I2C είναι το SDA (πράσινο) και το SCL (κίτρινο). Τα αντίστοιχα pins στο Arduino UNO είναι το Analog 4 και το Analog 5. Μπορούμε να τα συνδέσουμε απευθείας και δεν χρειάζεται κανένα εξάρτημα ενδιάμεσα. Τέλος, συνδέουμε το κόκκινο καλώδιο του βύσματος στο +5V του Arduino και το μαύρο στο GND. Αυτό ήταν όλο! Το Nunchuk έχει συνδεθεί πλέον με το Arduino και μπορούμε να ασχοληθούμε με την επικοινωνία τους. Πριν προχωρήσουμε, σημειώστε ότι αν η ιδέα του να κόψετε το βύσμα δεν σας αρέσει, υπάρχει μια πιο κομψή λύση, η οποία όμως κοστίζει. Συγκεκριμένα, υπάρχει μια μικρή πλακέτα που συνδέεται απευθείας στο Arduino UNO (σαν shield) και η οποία διαθέτει υποδοχή για το βύσμα του Nunchuk.

*Το Nunchuk (στα δεξιά) είναι ένα πρόσθετο του χειριστηρίου του Nintendo Wii. Για την επικοινωνία του χρησιμοποιεί το πρωτόκολλο I2C, το οποίο υποστηρίζεται *και* από το Arduino!*

Η δομή του software

Μετά από αρκετό ψάξιμο, μάθαμε ότι για να διαβάσουμε δεδομένα από τη συσκευή πρέπει προηγουμένως να γράψουμε κάποιες συγκεκριμένες τιμές, σε συγκεκριμένες θέσης της μνήμης του. Χρησιμοποιώντας τη βιβλιοθήκη Wire, πρέπει να γράψουμε την τιμή 0x55 στη θέση 0xF0 και την τιμή 0xFB στη θέση 0x00. Ουσιαστικά, αυτές οι τιμές αποτελούν μια ρύθμιση που επιβάλει στο Nunchuk να στέλνει αποκωδικοποιημένα τα δεδομένα του. Μάλιστα, αυτή η ρύθμιση ισχύει για κάθε χειριστήριο τύπου Nunchuk, από τα πρώτα που βγήκαν στην αγορά μέχρι τους πιο παράξενους κλώνους που διατίθενται στο eBay!

Για το χειρισμό του Nunchuk και την επικοινωνία μαζί του, δημιουργήσαμε μια κλάση με το (εμπνευσμένο) όνομα «Nunchuk». Μέσα σε αυτή την κλάση δημιουργήσαμε αρχικά μια μέθοδο, η οποία θα μπορεί να στέλνει bytes σε συγκεκριμένες θέσεις της μνήμης του Nunchuk. Στη συνέχεια γράψαμε μια άλλη μέθοδο, η οποία αρχικοποιεί τη συσκευή (με τη βοή-

2. Το datasheet περιέχει όλες τις τεχνικές πληροφορίες ενός εξαρτήματος: Από την τάση λειτουργίας και τις φυσικές του διαστάσεις, έως τον τρόπο με τον οποίο επικοινωνεί.

θεια της πρώτης μεθόδου) και διευθετεί μερικά ακόμα ζητήματα, όπως η επιλογή της συσκευής[3] και η απελευθέρωση του bus. Το όνομα που επιλέξαμε γι' αυτή τη μέθοδο είναι το «begin». Τέλος, για τη μεταφορά των δεδομένων, τη μετατροπή και την αποθήκευσή τους, δημιουργήσαμε μια ακόμα μέθοδο με το όνομα «update». Τα δεδομένα που στέλνει το χειριστήριο ταξιδεύουν με τη μορφή μιας εξάδας bytes. Ο ρόλος του κάθε byte φαίνεται στον ακόλουθο πίνακα.

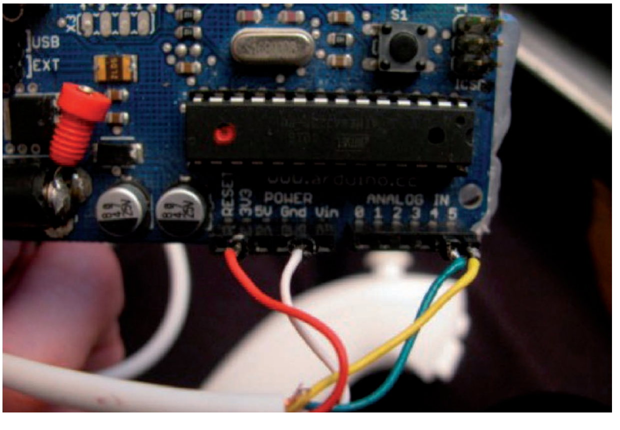
Byte #	Data type				
Byte 1	Joystick X				
Byte 2	Joystick Y				
Byte 3	Accelerometer X (bit 9 έως bit 2)				
Byte 4	Accelerometer Y (bit 9 έως bit 2)				
Byte 5	Accelerometer Z (bit 9 έως bit 2)				
Byte 6	Accel.Z bits 1:0	Accel.Y bits 1:0	Accel.X bits 1:0	C button	Z button

Όπως βλέπετε, τα πρώτα δύο bytes αναπαριστούν την οριζόντια και την κατακόρυφη μετατόπιση του joystick[4]. Η μετατόπιση της συσκευής ως προς κάθε άξονα του accelerometer (X, Y και Z) αναπαρίσταται από μια τιμή των 10bit. Τα επόμενα τρία bytes, λοιπόν, διαθέτουν τα 8 περισσότερα σημαντικά bits για καθεμία από αυτές τις τιμές. Στο έκτο byte βρίσκονται τα δύο λιγότερο σημαντικά bits για καθεμία από τις εν λόγω τιμές, καθώς και η κατάσταση των δύο πλήκτρων. Ας επιστρέψουμε τώρα στο πρόγραμμά μας. Η τρίτη συνάρτηση που γράψαμε παραλαμβάνει τα έξι bytes που στέλνει το Nunchuk, ξεμπλέκει τα δεδομένα κι αποθηκεύει κάθε τιμή σε μια ξεχωριστή μεταβλητή.

Υλοποίηση

Οι βιβλιοθήκες μπορούν να τοποθετηθούν σε διάφορες θέσεις. Υπάρχει όμως μία γενική θέση, εντός της οποίας τοποθετούνται οι βιβλιοθήκες με καθολική ισχύ[5]. Αυτή η θέση είναι ο φάκελος libraries, μέσα στον κατάλογο εγκατάστασης του Arduino IDE. Μια βιβλιοθήκη για το Arduino απαρτίζεται *τουλάχιστον* από ένα header file και από ένα αρχείο με κώδικα σε C. Στο αρχείο header δηλώνονται οι συναρτήσεις της βιβλιοθήκης, ενώ η υλοποίηση της καθεμίας βρίσκεται στο αρχείο με τον κώδικα. Παρεμπιπτόντως, το header file χρησιμοποιείται και για την ενσωμάτωση μιας βιβλιοθήκης σε ένα sketch (πρόκειται για το αρχείο που εισάγουμε στον κώδικά μας). Αν όλα αυτά σας φαίνονται γνώριμα, φταίει που με αυτόν ακριβώς τον τρόπο ορίζονται οι βιβλιοθήκες της C γενικότερα :)

Η σύνδεση μεταξύ Nunchuk και Arduino απαιτεί τέσσερα μόλις καλώδια. Δύο για τα δεδομένα του διαύλου I2C και δύο για την τροφοδοσία.



3. Σε έναν δίαυλο I2C μπορούν να συνδεθούν πολλές συσκευές ταυτόχρονα.
4. Το joystick είναι αναλογικό!
5. Δηλαδή αυτές που μπορούν να χρησιμοποιηθούν από όλα μας τα προγράμματα.

Σύμφωνα με όσα είπαμε, για να φτιάξουμε την πρώτη μας βιβλιοθήκη πρέπει να δημιουργήσουμε δύο αρχεία: Στο ένα θα βάλουμε τους ορισμούς των τριών μεθόδων που περιγράψαμε νωρίτερα, ενώ στο άλλο θα γράψουμε τον κώδικα που υλοποιεί κάθε μέθοδο. Στη συνέχεια θα τοποθετήσουμε τα δύο αρχεία σε έναν φάκελο, του οποίου το όνομα μπορούμε να επιλέξουμε αυθαίρετα. Φυσικά, καλό θα ήταν να επιλέξουμε ένα όνομα το οποίο δεν θα μας προκαλεί σύγχυση στο μέλλον. Τέλος, αυτός ο φάκελος θα τοποθετηθεί μέσα στον φάκελο libraries, εκεί όπου έχει εγκατασταθεί το Arduino IDE. Για τη συνέχεια, κατεβάστε κι αποσυμπιέστε το αρχείο, από το

http://bit.ly/dh019nunchuk



Αν δε θέλουμε να καταστρέψουμε το βύσμα του χειριστηρίου, μπορούμε να χρησιμοποιήσουμε το WiiChuck. Αυτή η μικρή πλακέτα επιτρέπει την απευθείας σύνδεση του βύσματος του Nunchuk επάνω της.

Λεπτομέρειες

Το περιεχόμενο των δύο αρχείων είναι απλό και θέλουμε να πιστεύουμε ότι δεν θα σας δυσκολέψει καθόλου. Ωστόσο, καλό θα ήταν να περιγράψουμε αναλυτικά το περιεχόμενου του καθενός, ώστε να βοηθηθούν και οι νεοφερμένοι! Ας τα εξετάσουμε με τη σειρά που τα γράψαμε.

Με το WiiChuck μπορούμε να τροφοδοτήσουμε το Nunchuk από τα analog pins του Arduino. Το χειριστήριο χρειάζεται λιγότερο ρεύμα από εκείνο που μπορούν να δώσουν τα pins του AVR και δεν υπάρχει κανένα πρόβλημα με αυτή τη διευθέτηση.

- **Nunchuk.h.** Πριν γράψουμε οτιδήποτε άλλο, ενσωματώνουμε τα αρχεία Arduino.h και Wire.h. Το πρώτο περιλαμβάνει τους απαραίτητους ορισμούς για τη λειτουργία ολόκληρου του περιβάλλοντος του Arduino, ενώ το άλλο αποτελεί τη βιβλιοθήκη που υποστηρίζει την επικοινωνία μέσω I2C. Επιπρόσθετα, ορίζουμε μια σταθερά με τη διεύθυνση I2C που χρησιμοποιεί το Nunchuk. Το κυρίως μέρος ξεκινά με τη δήλωση μιας κλάσης της C++, η οποία θα έχει το όνομα «NunchukClass». Στο δημόσιο (public) τμήμα της κλάσης, ορίζουμε τις μεταβλητές στις οποίες θα αποθηκεύονται τα δεδομένα που λαμβάνουμε από το Nunchuk. Στο ίδιο τμήμα της κλάσης ορίζουμε και τις μεθόδους begin και update. Στο ιδιωτικό (private) τμήμα της κλάσης ορίζουμε τη μέθοδο _sendByte, η οποία θα μπορεί να στέλνει μεμονωμένα bytes σε συγκεκριμένες θέσεις της μνήμης του Nunchuk.

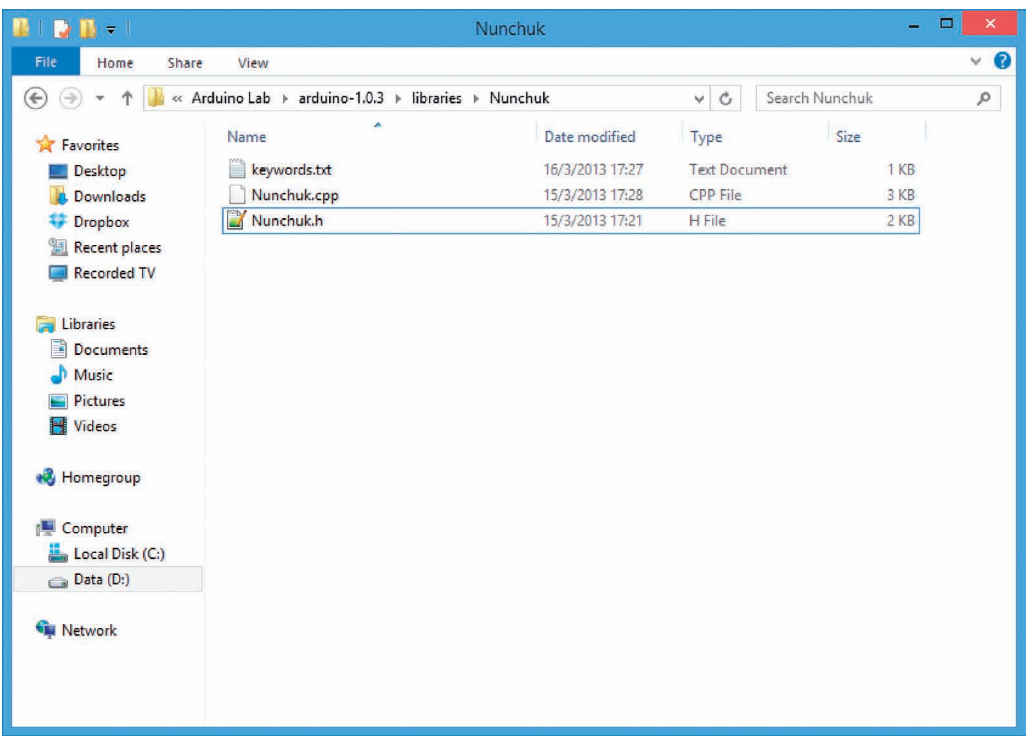
- **Nunchuk.cpp.** Εδώ πιστεύουμε ότι τα πράγματα είναι ακόμα πιο απλά. Κατ' αρχάς ξεκινάμε ενσωματώνοντας



το header file, το οποίο περιέχει τη δήλωση της κλάσης και των μεθόδων της. Κατόπιν γράφουμε τον κώδικα, που υλοποιεί τις τρεις μεθόδους. Ο κώδικας της μεθόδου begin ξεκινά ρυθμίζοντας κατάλληλα τους ακροδέκτες του Arduino, οι οποίοι συμμετέχουν στον δίαυλο I2C. Στη συνέχεια ο κώδικας φροντίζει για την αποστολή των δύο ειδικών bytes στη συσκευή, που τη ρυθμίζουν κατάλληλα ώστε να στέλνει τα δεδομένα της αποκωδικοποιημένα. Μετά κι απ' αυτό ο κώδικας συνεχίζει με μία κλήση της μεθόδου update, ώστε να γίνει η πρώτη ανάγνωση της συσκευής. Ο κώδικας για τη μέθοδο update αξιοποιεί τη συνάρτηση requestFrom από τη βιβλιοθήκη Wire, για να διαβάσει 6 bytes από το Nunchuk. Αφού ληφθούν τα δεδομένα ο κώδικας προχωρά με μία σειρά πράξεων bitwise, οι οποίες έχουν σαν στόχο το ξεκαθάρισμα των τιμών που στέλνει η συσκευή και την αποθήκευση της καθεμίας στην αντίστοιχη μεταβλητή. Προς το τέλος βρίσκεται και ο κώδικας για τη μέθοδο _sendByte. Εδώ χρησιμοποιούνται αποκλειστικά ορισμένες συναρτήσεις της βιβλιοθήκης Wire, οι οποίες αναλαμβάνουν την αποστολή δεδομένων προς τη συσκευή. Το Nunchuk.cpp τελειώνει με τη δήλωση ενός αντικειμένου της κλάσης NunchukClass. Αυτό γίνεται ώστε ο χρήστης να μπορεί άμεσα να μιλήσει με τη συσκευή, χρησιμοποιώντας το εν λόγω αντικείμενο (nunchcuk).

Η ώρα της δοκιμής

Όπως φαντάζεστε, μετά από τη συγγραφή του κώδικα της βιβλιοθήκης ακολούθησαν μερικές δοκιμές. Για το σκοπό αυτό κατασκευάσαμε ένα δοκιμαστικό πρόγραμμα, στο οποίο συμπεριλάβαμε τη νέα μας βιβλιοθήκη.



Το πρώτο αρχείο απευθύνεται στο Arduino IDE και φροντίζει για τον αυτόματο χρωματισμό των μεθόδων και των κλάσεων, από το σύστημα syntax highlighting. Τα άλλα δύο αρχεία περιέχουν τον κώδικα της βιβλιοθήκης.

```
void NunchukClass::begin(bool wiichuk) {
  Wire.begin();

  if (wiichuk) {
    pinMode(A2, OUTPUT); // Set analog pin 2 to output.
    pinMode(A3, OUTPUT); // Set analog pin 3 to output.
    digitalWrite(A2, LOW); // Set analog pin 3 to logical LOW.
    digitalWrite(A3, HIGH); // Set analog pin 3 to logical HIGH.
  }

  // http://wiibrew.org/wiki/Wiimote/Extension_Controllers#The_New_Way
  NunchukClass::_sendByte(0x55, 0xF0);
  NunchukClass::_sendByte(0x00, 0xFB);

  NunchukClass::update();
}
```

Η μέθοδος begin αρχικοποιεί τη βιβλιοθήκη Wire αλλά και το hardware. Για την περίπτωση που χρησιμοποιούμε το WiiChuck, συμπεριλάβαμε μια συνθήκη που ρυθμίζει κατάλληλα ορισμένα pins για την τροφοδοσία του χειριστηρίου.

Προφανώς, το συγκεκριμένο πρόγραμμα δεν χρειαζόταν να είναι μεγάλο, ούτε πολύπλοκο. Έτσι, το μόνο που έκανε ήταν να διαβάζει τα δεδομένα από το Nunchuk και να τα στέλνει στον υπολογιστή μέσω της σειραϊκής θύρας. Ας δούμε τις πρώτες γραμμές του.

```
#include <Wire.h>
#include <Nunchuk.h>

void setup() {
  Serial.begin(9600);
  nunchuk.begin();
}
```

Νομίζουμε ότι ο κώδικας μιλάει από μόνος του. Το πρόγραμμα ξεκινά με την ενσωμάτωση των βιβλιοθηκών Wire και Nunchuk. Εδώ χρειάζεται λίγη προσοχή: Η βιβλιοθήκη Wire χρησιμοποιείται από τη βιβλιοθήκη μας και γι' αυτό το λόγο πρέπει να ενσωματώνεται σε κάθε πρόγραμμα που τη χρησιμοποιεί! Ακολουθεί η γνώριμη συνάρτηση setup, στην οποία αρχικοποιούμε την επικοινωνία μέσω σειραϊκής θύρας, όπως και την επικοινωνία με το Nunchuk. Απλά και καθαρά πράγματα. Ας δούμε τι γίνεται στη συνέχεια.

```
void loop() {
  nunchuk.update();

  Serial.print(nunchuk.joystickX, DEC);
  Serial.print(" ");
  Serial.print(nunchuk.joystickY, DEC);
  Serial.print(" ");
  Serial.print(nunchuk.accelerometerX, DEC);
  Serial.print(" ");
  Serial.print(nunchuk.accelerometerY, DEC);
  Serial.print(" ");
  Serial.print(nunchuk.accelerometerZ, DEC);
  Serial.print(" ");
  Serial.print(nunchuk.buttonC, DEC);
  Serial.print(" ");
  Serial.print(nunchuk.buttonZ, DEC);
  Serial.println("");
}
```

Κατά καιρούς, η βιβλιοθήκη Wire υπόκειται σε ορισμένες αλλαγές. Για να αντιμετωπίσουμε τα προβλήματα που πηγάζουν από αυτές τις αλλαγές, ο κώδικας της βιβλιοθήκης ελέγχει την έκδοση του Arduino IDE και ενσωματώνει τα κατάλληλα κομμάτια κώδικα.

```
void NunchukClass::_sendByte(byte data, byte location) {
  Wire.beginTransmission(ADDRESS);

  #if defined(ARDUINO) && ARDUINO >= 100
  Wire.write(location);
  Wire.write(data);
  #else
  Wire.send(location);
  Wire.send(data);
  #endif

  Wire.endTransmission();
}
```

Αρχικά, η συνάρτηση loop καλεί τη μέθοδο update του αντικειμένου nunchuk. Έτσι, το πρόγραμμά μας λαμβάνει από τη συσκευή τις τιμές που περιγράφουν την κατάσταση των χειριστηρίων. Στη συνέχεια, με τη βοήθεια της μεθόδου print, το πρόγραμμα τυπώνει αυτές τις τιμές μέσω της σειραϊκής θύρας. Αυτό ήταν όλο! Περιττό να σας πούμε ότι το πρόγραμμά μας δούλεψε αμέσως και χωρίς το παραμικρό πρόβλημα. Φυσικά, πριν κάνουμε τη δοκιμή είχαμε φροντίσει να αντιγράψουμε τον φάκελο με τα αρχεία της βιβλιοθήκης στην προβλεπόμενη θέση (στον κατάλογο libraries, εκεί όπου έχει εγκατασταθεί το Arduino IDE).

Τελευταίες πινελιές

Μετά και από την προηγούμενη δοκιμή, βεβαιωθήκαμε: Η βιβλιοθήκη μας είναι έτοιμη και μπορούμε να την ενσωματώσουμε σε οποιαδήποτε εφαρμογή μας. Τι γίνεται όμως αν θελήσουμε να την ανεβάσουμε στο Internet, για να την αξιοποιήσουν κι άλλοι χρήστες του Arduino;

Πρώτα απ' όλα θα χρειαστούμε μια άδεια χρήσης. Η διασημότερη και πιο διαδεδομένη άδεια είναι η GPL. Μπορούμε φυσικά να διαλέξουμε οποιαδήποτε άλλη, αλλά εμείς προτιμάμε αυτήν! Το μόνο που χρειάζεται να

```
void NunchukClass::update() {
  int count = 0;
  int values[6];

  Wire.requestFrom(ADDRESS, 6);

  while(Wire.available()) {
    values[count] = Wire.read();
    count++;
  }

  NunchukClass::joystickX = values[0];
  NunchukClass::joystickY = values[1];
  NunchukClass::accelerometerX = (values[2] << 2) | ((values[5] >> 2) & 3);
  NunchukClass::accelerometerY = (values[3] << 2) | ((values[5] >> 4) & 3);
  NunchukClass::accelerometerZ = (values[4] << 2) | ((values[5] >> 6) & 3);
  NunchukClass::buttonZ = !((values[5] >> 0) & 1);
  NunchukClass::buttonC = !((values[5] >> 1) & 1);

  NunchukClass::_sendByte(0x00, 0x00);
}
```

Μεταξύ άλλων, η μέθοδος update «σπάει» τα έξι 6 bytes που στέλνει το Nunchuk σε επτά μεταβλητές κι έτσι διευκολύνει την αξιοποίησή τους από το υπόλοιπο πρόγραμμα.

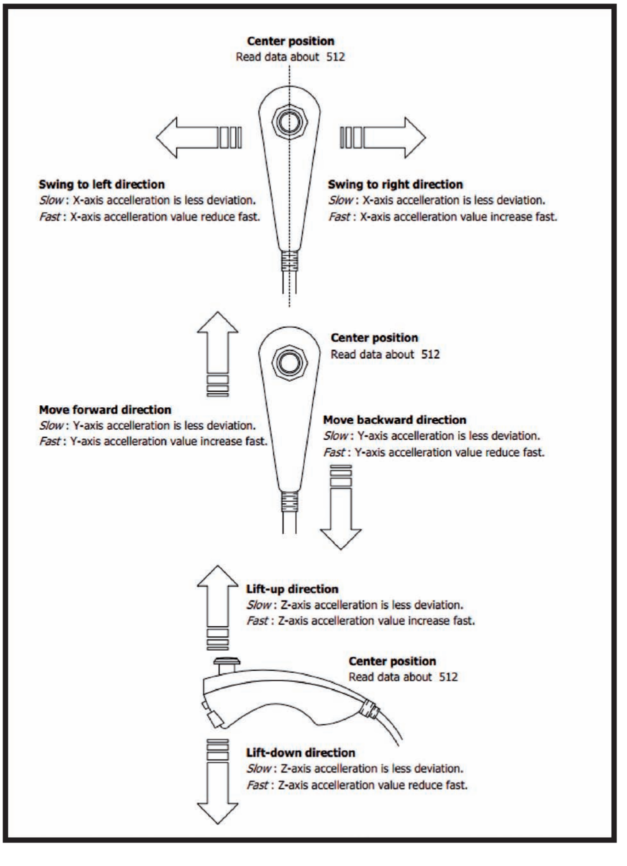
κάνουμε είναι να συμπεριλάβουμε σε κάθε αρχείο μια μικρή παράγραφο, η οποία θα δηλώνει ότι ο κώδικάς μας καλύπτεται από τη συγκεκριμένη άδεια. Προαιρετικά, μπορούμε να δίνουμε μαζί με τα αρχεία της βιβλιοθήκης κι ένα αντίγραφο της άδειας. Πέρα από την άδεια χρήσης, θα χρειαστεί να φτιάξουμε κι ένα αρχείο με το όνομα keywords.txt. Αυτό το αρχείο χρησιμοποιείται από το Arduino IDE για τον χρωματισμό των μεθόδων και των αντικειμένων της βιβλιοθήκης, από το σύστημα syntax highlighting του Arduino IDE. Το αρχείο μας θα πρέπει να έχει την εξής μορφή:

```
Nunchuk KEYWORD1
begin         KEYWORD2
update        KEYWORD2
```

Ουσιαστικά, μέσα στο συγκεκριμένο αρχείο δηλώνουμε τα ονόματα των μεθόδων και των αντικειμένων της βιβλιοθήκης, τα οποία θέλουμε να χρωματίζονται από το Arduino IDE. Προσέξτε ότι το Nunchuk δηλώνεται σαν KEYWORD1, που σημαίνει ότι πρόκειται για αντικείμενο. Τα begin και update δηλώθηκαν σαν KEYWORD2, που σημαίνει ότι αποτελούν ονόματα μεθόδων.

Τέλος, για να διευρύνουμε το πεδίο εφαρμογών της βιβλιοθήκης μας, φροντίσαμε κάτι ακόμα. Στην αρχή του άρθρου αναφερθήκαμε στο WiiChuk, τη μικρή πλακέτα που συνδέεται στο Arduino και παρέχει μια υποδοχή για τη σύνδεση του Nunchuk. Η εν λόγω πλακέτα συνδέεται σε τέσσερα αναλογικά pins του Arduino board. Τα δύο από αυτά ανήκουν στο δίαυλο I2C, ενώ τα άλλα δύο χρησιμοποιούνται για την τροφοδοσία. Όπως αντιλαμβάνεστε, τα pins που προορίζονται για την τροφοδοσία του WiiChuk πρέπει να είναι ρυθμισμένα κατάλληλα (το ένα HIGH και το άλλο LOW, ώστε να δίνουν 5V και 0V αντίστοιχα). Ε, λοιπόν, αυτό το ζητηματάκι φροντίσαμε να το αντιμετωπίσουμε στον κώδικα της μεθόδου begin! Αν μελετήσετε τον κώδικα, θα δείτε ότι η συγκεκριμένη μέθοδος δέχεται μία παράμετρο. Με αυτήν καθορίζουμε το αν πρόκειται να χρησιμοποιηθεί το WiiChuk και άρα το αν θέλουμε να τροφοδοτηθεί από τα αντίστοιχα analog pins.

Αυτή την εικόνα τη βρήκαμε στο Διαδίκτυο. Μας βοηθά να καταλάβουμε τι σημαίνουν οι τιμές που στέλνει το χειριστήριο, καθώς και πώς μπορούμε να τις μεταφράσουμε σε κάτι πιο χρήσιμο...





Ο γείτονας που κανείς δεν ξέρει, αλλά όλοι αγαπάνε!

Skill: Intermediate

Tags: Captive, Squid, Portals, Wireless Access Points, Pineapple, pfSense, proxy cache, ntop, packet capturing

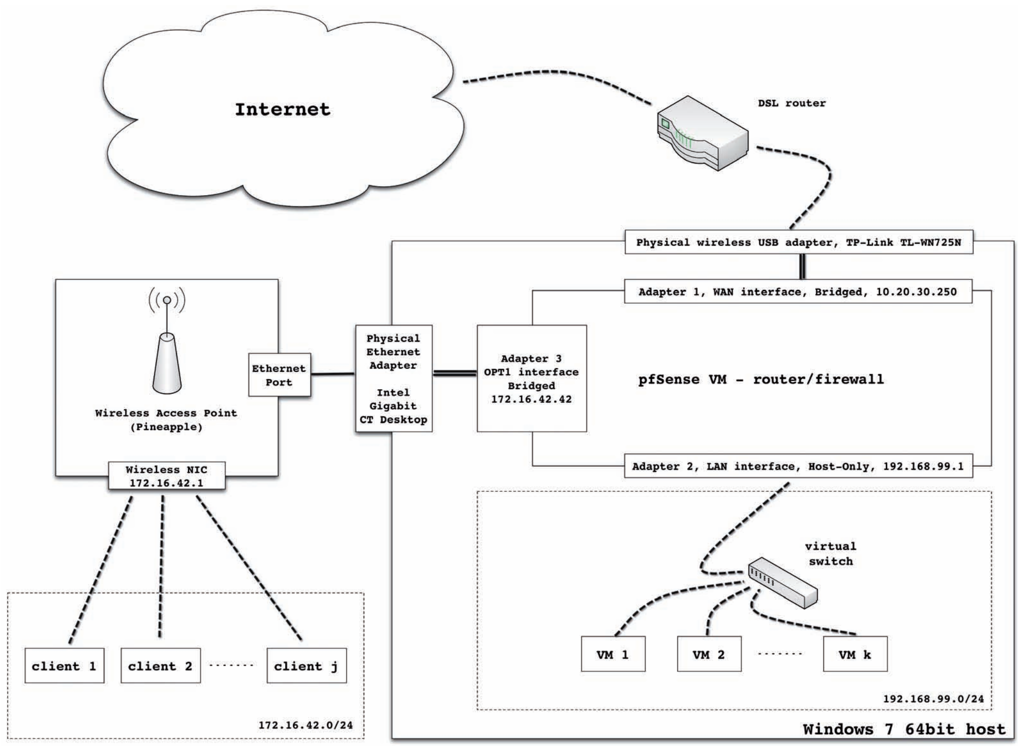
Εκείνον που αναζητά ανοικτά, ιδιωτικά δίκτυα WiFi, τον γνωρίζετε. Ξέρετε επίσης κι αυτόν που ψάχνει ασύρματα δίκτυα με αδύναμη ασφάλεια, συχνά μόνο και μόνο για τη χαρά του σπασίματος. Δεν υπονοούμε ότι είναι φίλοι σας, ούτε ότι κάνετε εσείς τέτοια πράγματα — προς Θεού, δηλαδή. Σίγουρα όμως θα έχετε υπ' όψιν ότι κάτι τέτοιοι τύποι κυκλοφορούν Εκεί Έξω (TM).

by sub2raw

Αντί στο παρόν άρθρο να συζητήσουμε για το πώς βρίσκουν οι φίλοι μας ασύρματα δίκτυα, πώς επιχειρούν να παίρνουν πρόσβαση και υπό ποίες προϋποθέσεις το καταφέρνουν, λέμε να κάνουμε κάτι *ελαφρώς* διαφορετικό: Να δείξουμε πώς μοιράζουμε απλόχερα κι ασύρματα Ίντερνετ, σ' ολόκληρη τη γειτονιά. Από την αρχή σας λέμε ότι δεν έχουμε κάτι απλοϊκό κατά νου, στο στιλ «σετάρω ένα ανοικτό wireless access point και ξεχνάω ότι υπάρχει και λειτουργεί». Οι ιδέες που υλοποιήσαμε και σε λίγο θα σας παρουσιάσουμε είναι λίγο πιο περίπλοκες — αλλά και πιο ενδιαφέρουσες. Είναι βέβαια λογικό ν' αναρωτιέστε: Γιατί να μοιράσω Ίντερνετ στη γειτονιά; Ακολουθούν μερικοί μόνο από τους λόγους που μπορούμε να σκεφτούμε, όχι κατ' ανάγκη σε σειρά σημαντικότητας.

- Δεν σας χαλάει να μοιράζεστε κάτι, το οποίο διαθέτετε σε αφθονία (VDSL, κανείς;)
- Έχετε πέσει πάνω σε Captive Portal —ή έστω έχετε ακούσει γι' αυτά— και πάντα θέλατε να στήσετε το δικό σας.
- Ψάχνετε για έναν πρωτότυπο τρόπο να διαφημίσετε λίγο το εταιρικό ή ακόμη και το προσωπικό σας site. (Θα δείτε τι εννοούμε αργότερα.)
- Θέλετε να πειραματιστείτε στην πράξη με proxies, firewalls και τα συναφή. Το έχουμε πει πολλές φορές και δεν θα βαρεθούμε να το επαναλαμβάνουμε: Σαν την πρακτική εξάσκηση, δεν έχει!
- Σκεφτόσαστε ότι καλό θα ήταν να περάσετε ένα ή περισσότερα ευχάριστα απογεύματα — ενδεχομένως λίγο πιο ευχάριστα από εκείνα που συχνά μάς αναφέρει μέσω twitter ο αγαπητός συνεργάτης και φίλος, @sonic2000gr.

Σχηματική αναπαράσταση του δικτύου που στήσαμε, προκειμένου να παρέχουμε ασύρματη πρόσβαση στη γειτονιά και να μοιράζουμε την ιντερνετική μας σύνδεση.



- Έχετε ακούσει ότι πίσω από κάποια wireless access points γίνονται διάφορα παράξενα και περίεργα πράγματα, οπότε τώρα σκέφτεστε να διαπιστώσετε στην πράξη τι δυνατότητες υπάρχουν.
- Διαβάσατε το άρθρο του Ion, που αρχίζει από τη σελίδα 22 του παρόντος τεύχους. Σας έχουν μπει στο μυαλό ιδέες τις οποίες οπωσδήποτε πρέπει να δοκιμάσετε.
- Θέλετε ν' αποδείξετε στον ξερόλα γείτονα ότι δεν είναι τόσο ασφαλής στο Ίντερνετ — τουλάχιστον όχι τόσο, όσο νομίζει.
- Σκοπεύετε να κάνετε κάποιες αλλαγές στο γραφείο σας ή στη εταιρεία. Θα ήταν καλά να δίνατε σε (υποψήφιους) πελάτες ή/και σε επισκέπτες ελεύθερη πρόσβαση στο Ίντερνετ, κρατώντας τους όμως μακρυά από το τοπικό δίκτυο.
- Σχεδιάζετε για τους μαθητές σας μερικά πρακτικά μαθήματα περί δικτύων ή/και ασφάλειας. Δεν χρειάζεται να σας πούμε πόσο «επικίνδυνος» μπορεί να γίνει με τις ερωτήσεις του ένας μαθητής που διακατέχεται από γνήσια περιέργεια. Πριν λοιπόν ξεκινήσετε τα μαθήματα, ας είστε όσο καλύτερα προετοιμασμένος γίνεται.

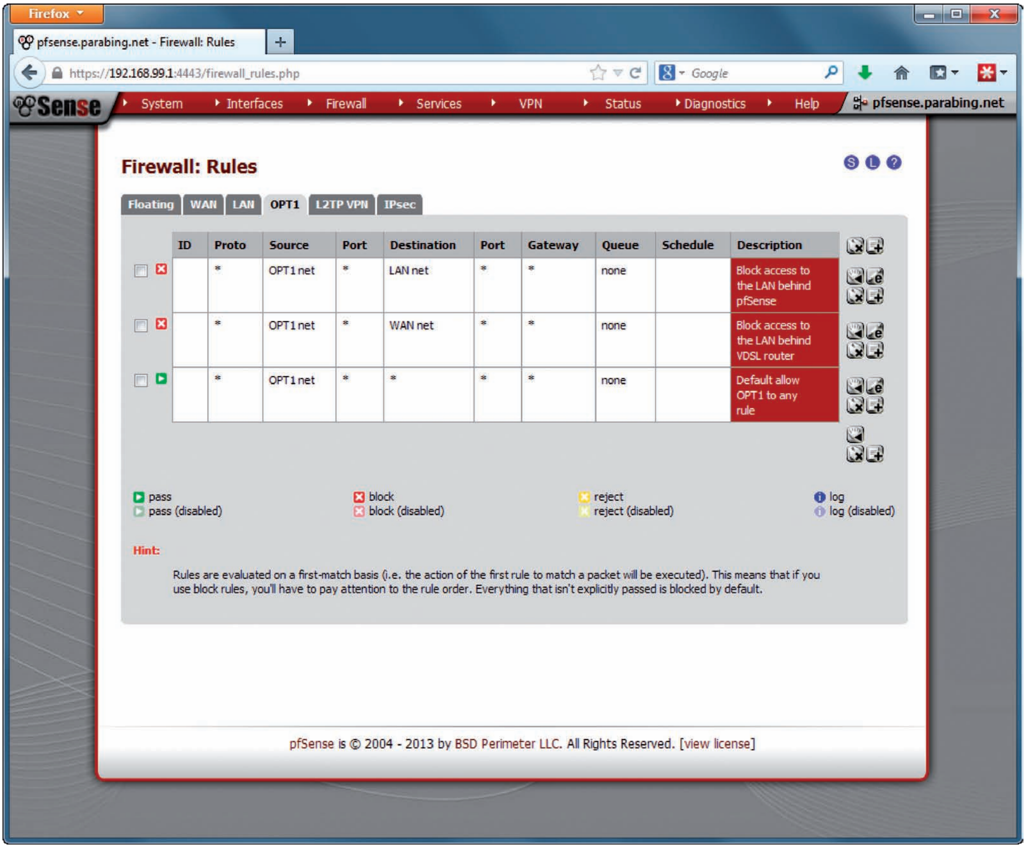
Θα μπορούσαμε να συνεχίσουμε με τη λίστα, πιστεύουμε ωστόσο ότι τα κίνητρά μας έχουν ήδη γίνει φανερά και κατανοητά. Χωρίς άλλες περιστροφές, λοιπόν, ας περάσουμε στο προκείμενο.

Τι είν' αυτά τα Captive Portals;

Με απλά λόγια, είναι σελίδες HTML στις οποίες κατευθυνόμαστε αυτομάτως την πρώτη φορά που επιχειρούμε να χρησιμοποιήσουμε κάποιο WiFi HotSpot ή κάποια ενσύρματη σύνδεση, π.χ., σε εταιρεία ή ξενοδοχείο. Πριν αποκτήσουμε πρόσβαση στο Ίντερνετ, ουσιαστικά είμαστε μπλοκαρισμένοι από το firewall πίσω από το Captive Portal. Αφού ανοίξουμε έναν οποιονδήποτε web browser και προσπαθήσουμε να επισκεφτούμε έναν δικτυακό τόπο, αντί γι' αυτόν θα δούμε τη σελίδα του Captive Portal. Αναλόγως της υλοποίησης, είναι πιθανό να μας ζητείται login, να πληρώσουμε με χρήση πιστωτικής ή να δώσουμε έναν κωδικό που αναγράφεται, π.χ., σε μια απόδειξη που μόλις πήραμε. Εναλλακτικά, η σελίδα του Captive Portal μπορεί απλά να παραθέτει τους όρους χρήσης της δικτυακής υπηρεσίας που πρόκειται να χρησιμοποιήσουμε. Μόλις κάνουμε την απαιτούμενη ενέργεια —login, πληρωμή, πληκτρολόγηση κωδικού ή απλά αποδοχή των όρων χρήσης—, οι περιορισμοί του firewall αίρονται και μπορούμε, πλέον, να χρησιμοποιούμε το Ίντερνετ κανονικά, ενδεχομένως για ένα προκαθορισμένο χρονικό διάστημα.

Τι έχουμε, τι θέλουμε

Είναι προφανές ότι διαθέτουμε ήδη μια σύνδεση στο Ίντερνετ. Σκοπεύουμε να δώσουμε σε επισκέπτες, γείτονες, περαστικούς κ.ά. την ευκαιρία να τη χρησιμοποιούν κι εκείνοι, ασύρματα. Την ίδια στιγμή, όλοι αυτοί οι χρήστες θέλουμε να είναι απομονωμένοι από το δικό μας δίκτυο, που βρίσκεται πίσω από τον router. Προσέξτε: Δεν επιθυμούμε απλά να είναι σε άλλο υποδίκτυο, αλλά να μη μπορούν καν να βλέπουν τους δικούς μας υπολογιστές, τις ασύρματες ή ενσύρματες συσκευές — ούτε τον ίδιο τον router.



Όσοι συνδέονται στο ανοικτό wireless access point μας δεν θέλουμε να έχουν πλήρη πρόσβαση στο Ίντερνετ αλλά να μη βλέπουν το φυσικό, τοπικό μας δίκτυο, ούτε και το εικονικό δίκτυο πίσω από το pfSense. Οι wireless clients προέρχονται από το OPT1 net, το φυσικό τοπικό δίκτυο για το pfSense είναι το WAN net, ενώ οι εικονικοί υπολογιστές πίσω από το pfSense βρίσκονται στο LAN net. Μετά απ' αυτά, οι επιθυμητοί κανόνες για το firewall συγκροτούνται πανεύκολα.

Τώρα, από πλευράς hardware, αν ο router μας διαθέτει wireless interface και παρέχει λειτουργία Captive Portal, είμαστε μια χαρά. Για παράδειγμα, στην περίπτωση που ο router μπορεί να πάρει dd-wrt ή OpenWrt firmware και του το έχουμε περάσει, αυτομάτως αποκτάμε και τη λειτουργία Captive Portal ή αλλιώς HotSpot[1]. Διαφορετικά, θα χρειαστούμε κατ' ελάχιστον είτε ένα αυτόνομο wireless access point (WAP) είτε μια κάρτα ασύρματης δικτύωσης, η οποία όμως μπορεί και μπαίνει σε λειτουργία access point (AP). Φυσικά, η κάρτα αυτή θα πρέπει να υποστηρίζεται πλήρως κι από το OS, στο πλαίσιο του οποίου θα λειτουργεί ο router για τους επισκέπτες.

Πιθανώς κάπου εδώ τα πράγματα να θόλωσαν. Μην ανησυχείτε, το φταίξιμο είναι αποκλειστικά δικό μας. Σε πολύ λίγο, που θα σας έχουμε περιγράψει τι κάναμε για τις ανάγκες των δικών μας πειραματισμών, θα ξεκαθαρίσουν και πάλι.

Το δικό μας setup

Ως WAP μπορούμε να επιλέξουμε μια φθηνή, χωρίς ιδιαίτερες δυνατότητες συσκευή. Εμείς είχαμε ένα Pineapple Mark 3, το οποίο απεδείχθη μια χαρά για τους σκοπούς μας.

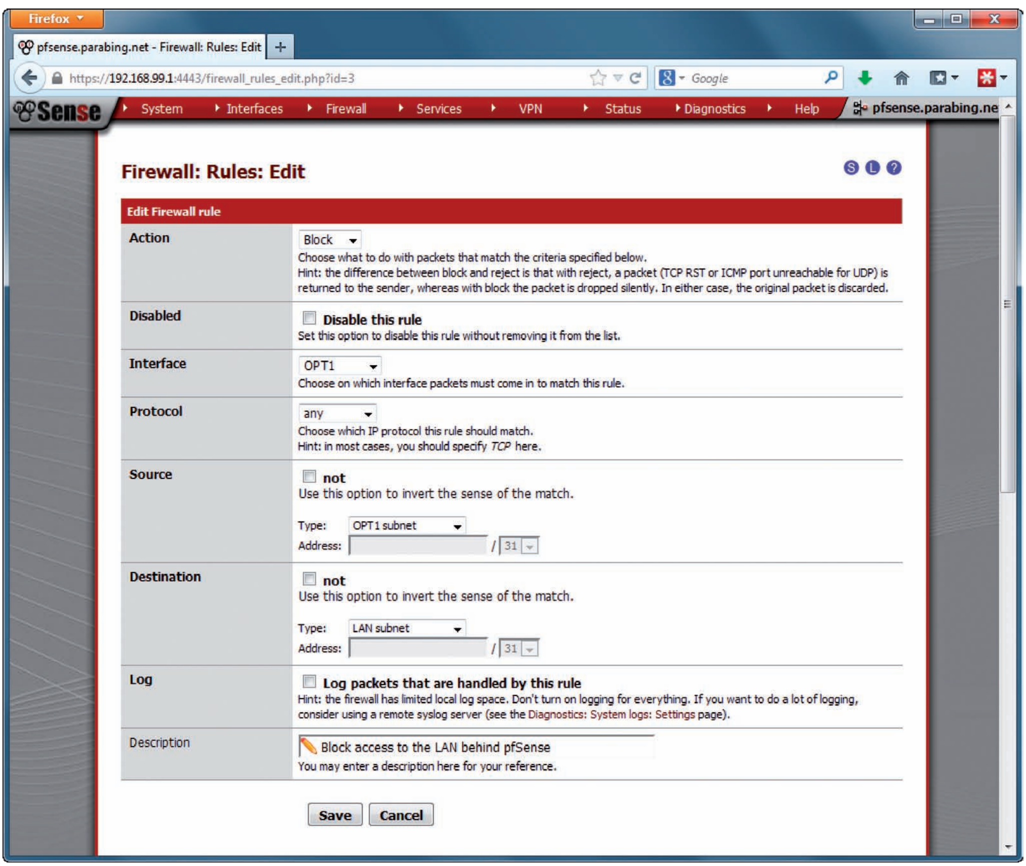
1. Περί dd-wrt και OpenWrt μπορείτε να διαβάσετε στα σχετικά άρθρα που φιλοξενούμε στα τεύχη 011, 012 και 015.

Pineapple: Ένα ιδιαίτερα πρόθυμο access point

Το Pineapple είναι ένας wireless router τον οποίο οποιοσδήποτε μπορεί να προμηθευτεί από το HakShop (`http://hakshop.myshopify.com`). Η βασική του λειτουργία είναι να προσφέρει απλόχερα ασύρματη πρόσβαση σε υπολογιστές, laptops, smartphones και άλλες δικτυακές συσκευές, παριστάνοντας γνωστά γι' αυτές Access Points! Έτσι, επιτρέπει στον κάτοχό του να εξαπολύει επιθέσεις man in the middle (MiTM) προς όσους το χρησιμοποιούν. Εναλλακτικά, μπορεί να χρησιμοποιηθεί κι ως ένα απλό wireless access point. Περισσότερα για τη λειτουργία του Pineapple μπορείτε να διαβάσετε στο δικτυακό του τόπο αλλά και στο αναλυτικό άρθρο που φιλοξενούμε στο deltaHacker 005, όπου θα μάθετε και τις ιδιαιτερότητες της ασύρματης δικτύωσης που εκμεταλλεύεται η συσκευή.

Γενικά, το WAP πρέπει να συνδεθεί πάνω σε κάποιον router, ο οποίος ναι μεν θα παρέχει πρόσβαση στο Ίντερνετ στους wireless clients, ταυτόχρονα όμως θα τους απομονώνει από το υπόλοιπο δίκτυο πίσω του. Αντί για τον VDSL router στο δικό μας τοπικό δίκτυο, στραφήκαμε σε μια εγκατάσταση του pfSense. Δύο ήταν οι βασικοί λόγοι: Αφενός θέλαμε απόλυτο έλεγχο πάνω στο νέο (ασύρματο) δίκτυο που θα φτιάχναμε,

Ο ορισμός του κανόνα που εμποδίζει τους wireless clients να βλέπουν τις εικονικές μηχανές πίσω από το pfSense. Ο κανόνας είναι σαν να λέει: "Μπλόκαρε τα πακέτα που προέρχονται από το OPT1 subnet του OPT1 interface και προορίζονται για το LAN subnet. Α, καθόλου δεν μ' ενδιαφέρει το πρωτόκολλο, το source port ή το destination port."



αφετέρου σκοπεύαμε να εγκαταστήσουμε εφαρμογές κι εργαλεία που θα βελτίωναν την εμπειρία των wireless clients — αλλά θα παρείχαν και σ' εμάς μερικές εξαιρετικά ενδιαφέρουσες δυνατότητες.

pfSense: O router που δεν μπορείτε να μην αγαπήσετε!

Το pfSense είναι ένα δωρεάν, πανίσχυρο λειτουργικό σύστημα που προορίζεται για χρήση firewall/router (`http://pfsense.org`). Βασίζεται στο FreeBSD (`http://freebsd.org`) και χρησιμοποιεί το pf firewall, το οποίο ξεκίνησε από το OpenBSD (`http://openbsd.org`).

Στις πολλές δυνατότητες του pfSense συμπεριλαμβάνονται το load balancing, το traffic shaping, το packet sniffing και πολλές άλλες, όπως η υποστήριξη dynamic DNS, VPN διαφορετικών τύπων και η λειτουργία Captive Portal. Για το pfSense υπάρχει μια πληθώρα έτοιμων εφαρμογών, οι οποίες μπορούν να προστεθούν εκ των υστέρων και να επαυξήσουν τις δυνατότητες του λειτουργικού. Ενδεικτικά αναφέρουμε συστήματα Intrusion Detection και web proxies. Τα μηχανήματα που βρίσκονται πίσω από το pfSense...

- παίρνουν διευθύνσεις IP από τον DHCP server
- χρησιμοποιούν τον DNS server του pfSense
- επικοινωνούν με τα μηχανήματα του έξω κόσμου...
- ...αλλά προστατεύονται κι απ' αυτά, χάρη στο firewall του pfSense

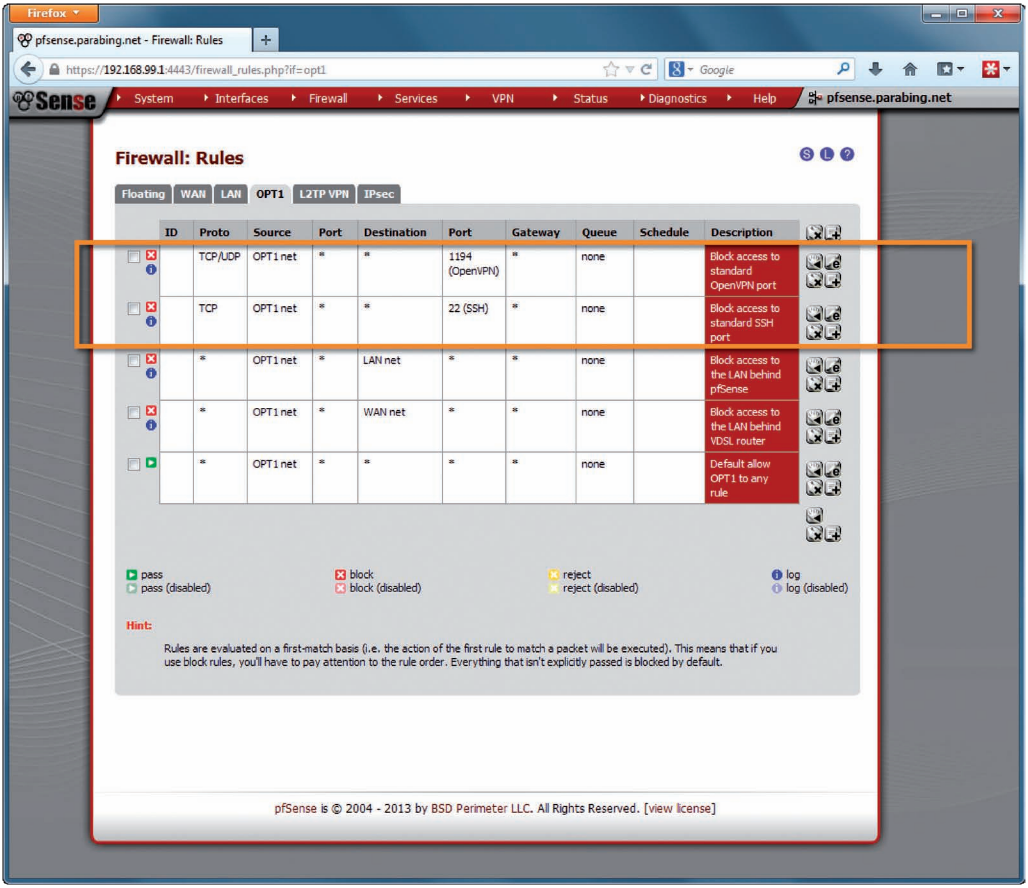
Παρακολουθήστε μια λεπτομερή παρουσίαση της εγκατάστασης, της ρύθμισης αλλά και της χρήσης του pfSense στο 7ο επεισόδιο της 1ης σεζόν του deltaCast:

`http://deltahacker.gr/deltacast-s01e07`

Στο παρόν άρθρο θεωρούμε το pfSense ήδη εγκατεστημένο και τον χρήστη του εξοικειωμένο μ' αυτό.

Η εγκατάσταση του pfSense δεν έγινε σε φυσικό μηχάνημα αλλά σε virtual machine (VM) του VirtualBox, το οποίο με τη σειρά του ήταν ήδη εγκατεστημένο σ' ένα τυπικό PC με host OS τα Windows 7 64bit. Ενδέχεται να φαντάζει λίγο παράξενο αυτό το setup. Σε περίπτωση όμως που δεν έχουμε το hardware για ν' αφιερώσουμε αποκλειστικά στο pfSense, είναι περισσότερο από μια χαρά. Το VM με το pfSense είχε τρία network adapters.

- **Adapter 1, Bridged.** Το WAN interface του pfSense. Από αυτό, το λειτουργικό έβλεπε απευθείας τον (φυσικό) router του τοπικού δικτύου κι έβγαινε στο Ίντερνετ. Η συσκευή του host computer στην οποία ήταν αντιστοιχισμένο (attached) το εν λόγω bridged interface, ήταν ένας μικροσκοπικός wireless USB 2.0 adapter της TP-Link (TL-WN725N).
- **Adapter 2, Host-only.** Το LAN interface του pfSense. Αυτό βλέπουν όσα VMs βρίσκονται πίσω από τον εικονικό firewall/router. Τα συγκεκριμένα VMs βγαίνουν στον έξω κόσμο μέσω του pfSense. Σ' αυτό το δίκτυο θα μπορούσαμε να έχουμε ένα VM με το Kali Linux — αλλά αδυνατούμε να φανταστούμε έναν πρακτικό λόγο για να θέλουμε κάτι παρόμοιο :)



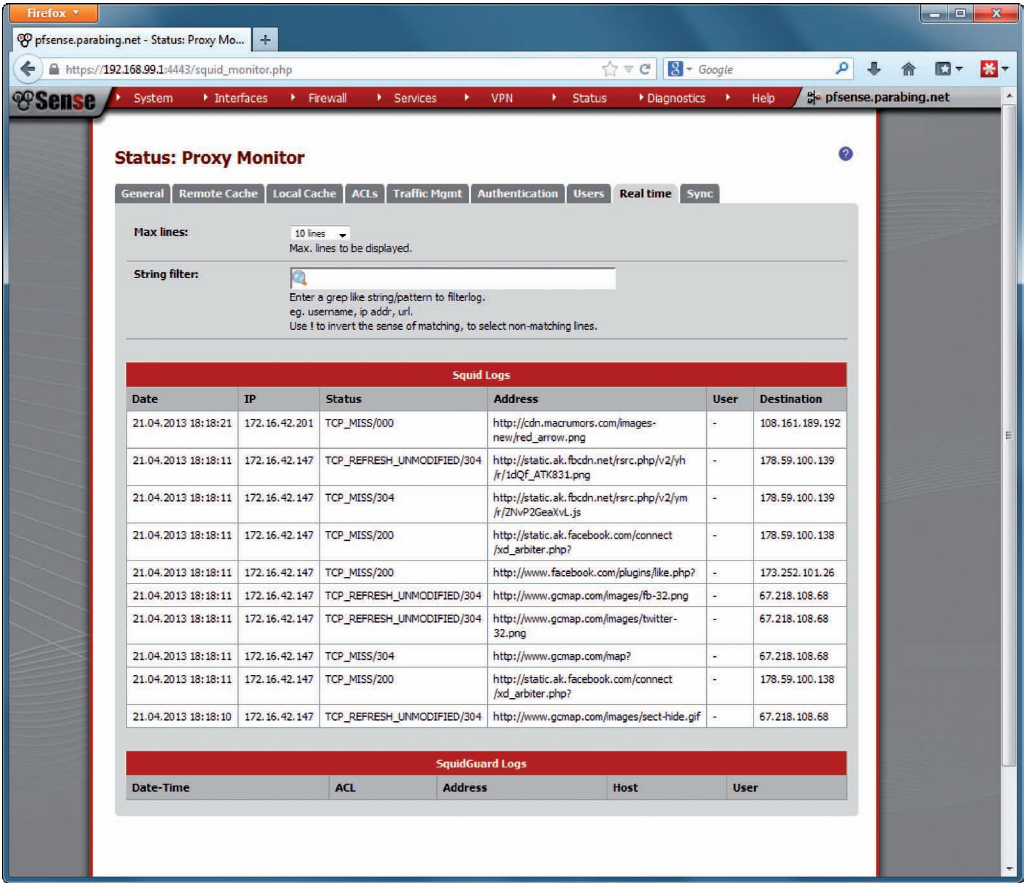
- **Adapter 3, Bridged.** Το OPT1 interface του pfSense, το οποίο ήταν bridged με την κάρτα Ethernet του PC (ένας Intel Gigabit CT Desktop Adapter). Πάνω στο (φυσικό) Ethernet port συνδέεται, μέσω καλωδίου UTP, το εκάστοτε WAP. Εμείς, π.χ., είχαμε συνδέσει εκεί το Pineapple.

Αφού φροντίσουμε για όλα τα προηγούμενα και ρυθμίσουμε το WAP ώστε να παρέχει ανοικτή πρόσβαση (δηλαδή χωρίς WEP, WPA κ.λπ.), αυτομάτως γινόμαστε υποψήφιοι για τον τίτλο του πιο γενναιόδωρου γείτονα! Χρειάζονται βέβαια και κάποιες μικρορυθμίσεις στο firewall μπροστά από το WAP (εν προκειμένω σ' εκείνο του pfSense), ενώ δεν έχουμε ενεργοποιήσει ακόμη και τη λειτουργία Captive Portal. Αυτά κι άλλα πολλά (το εννοούμε) θα τα δούμε με λεπτομέρειες στη συνέχεια του άρθρου.

Βασική ρύθμιση του firewall

Οι clients που συνδέονται πάνω στο wireless access point μας (στο Pineapple), θέλουμε να έχουν ανεμπόδιστη πρόσβαση παντού *εκτός* από δύο τοπικά δίκτυα: Εκείνο πίσω από το LAN interface του pfSense κι εκείνο πίσω από τον (φυσικό) DSL router.

Εδώ ο ιδιοκτήτης του ανοικτού wireless access point αποφασίζει να δυσκολέψει τη ζωή των χρηστών του, προσθέτοντας στο firewall δύο κανόνες που απαγορεύουν τις συνδέσεις σε SSH και OpenVPN servers. Πλέον, η μόνη ασφάλεια των clients παρέχεται από το πρωτόκολλο HTTPS. Αν βέβαια έχουν μεριμνήσει, οι SSH/OpenVPN servers που διαθέτουν δεν χρησιμοποιούν τυπικά ports. Έτσι, οι περιοριστικοί κανόνες του firewall δεν τους αφορούν.



Date	IP	Status	Address	User	Destination
21.04.2013 18:18:21	172.16.42.201	TCP_MISS/000	http://cdn.macrumors.com/images-new/red_arrow.png	-	108.161.189.192
21.04.2013 18:18:11	172.16.42.147	TCP_REFRESH_UNMODIFIED/304	http://static.ak.fbcdn.net/rsrc.php/v2/yh/r/1dQf_ATK831.png	-	178.59.100.139
21.04.2013 18:18:11	172.16.42.147	TCP_MISS/304	http://static.ak.fbcdn.net/rsrc.php/v2/ym/r/ZNvP2GeaXvL.js	-	178.59.100.139
21.04.2013 18:18:11	172.16.42.147	TCP_MISS/200	http://static.ak.facebook.com/connect/xd_arbiter.php?	-	178.59.100.138
21.04.2013 18:18:11	172.16.42.147	TCP_MISS/200	http://www.facebook.com/plugins/like.php?	-	173.252.101.26
21.04.2013 18:18:11	172.16.42.147	TCP_REFRESH_UNMODIFIED/304	http://www.gcmap.com/images/fb-32.png	-	67.218.108.68
21.04.2013 18:18:11	172.16.42.147	TCP_REFRESH_UNMODIFIED/304	http://www.gcmap.com/images/twitter-32.png	-	67.218.108.68
21.04.2013 18:18:11	172.16.42.147	TCP_MISS/304	http://www.gcmap.com/map?	-	67.218.108.68
21.04.2013 18:18:11	172.16.42.147	TCP_MISS/200	http://static.ak.facebook.com/connect/xd_arbiter.php?	-	178.59.100.138
21.04.2013 18:18:10	172.16.42.147	TCP_REFRESH_UNMODIFIED/304	http://www.gcmap.com/images/sect-hide.gif	-	67.218.108.68

Ο proxy server που έχει ο κάτοχος του wireless access point επιταχύνει την εμπειρία web surfing των clients, αλλά υπάρχει και η άλλη όψη του νομίσματος: Οι περιηγήσεις τους στο web παρακολουθούνται εύκολα και μάλιστα σε πραγματικό χρόνο! Πιστεύουμε ότι η ανάγκη χρήσης OpenVPN, ειδικά όταν σερφάρουμε από δημόσια HotSpots, γίνεται πλέον προφανής.

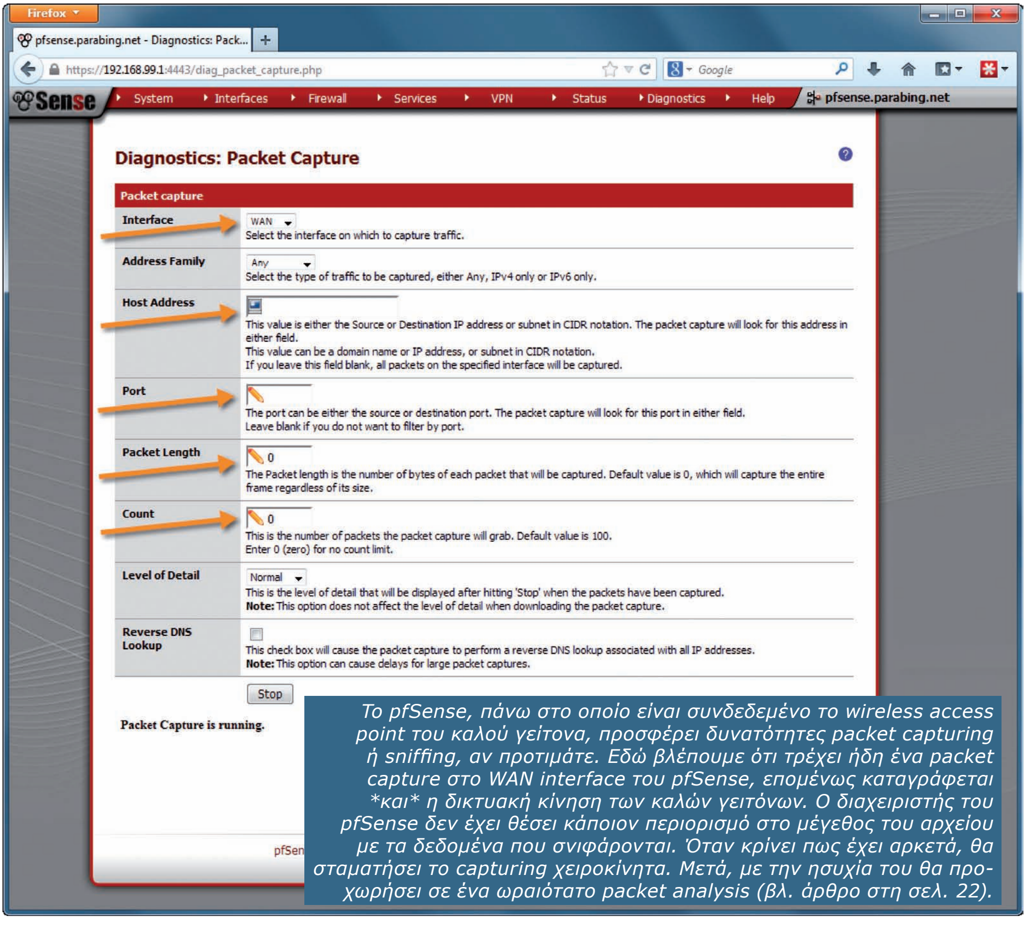
Προς τούτο, χρειάζονται δύο σχετικοί κανόνες για το OPT1 interface του pfSense. Αναλυτικότερα, όποιο πακέτο προέρχεται από το OPT1 net (δηλαδή από κάποιον client του Pineapple) κι έχει προορισμό το LAN net (το δίκτυο πίσω από το pfSense) ή το WAN net (το δίκτυο πίσω από τον DSL router), θα μπλοκάρεται. Οι κανόνες αυτοί ορίζονται εύκολα από το web panel του pfSense, ακολουθώντας τη διαδρομή Firewall → Rules → OPT1.

Ρύθμιση κι ενεργοποίηση του Captive Portal

Όσοι σκοπεύουν να χρησιμοποιήσουν το ελεύθερο wireless access point μας είναι ευπρόσδεκτοι, πρέπει ωστόσο να γνωρίζουν πού πάνε να μπλέξουν :) Επιπρόσθετα, αφού αποδεχθούν κάποιους βασικούς όρους χρήσης και πριν αρχίσουν το δωρεάν surfing, καθόλου δεν θα έβλαπτε αν έβλεπαν στον web browser μια ιστοσελίδα που εμείς θα είχαμε καθορίσει. Στη γειτονιά της Parabing Creations, για παράδειγμα, οι χρήστες του ανοικτού access point που έχουμε στήσει πριν ξεκινήσουν την κανονική περιήγηση κατευθύνονται στο `http://deltahacker.gr`. Από εκεί και πέρα, εννοείται ότι μπορούν να πηγαίνουν (σχεδόν) όπου θέλουν.

Από το web panel του pfSense, η ρύθμιση της υπηρεσίας Captive Portal γίνεται από το Services → Captive Portal. Στην ομώνυμη καρτέλα, μετά

από ένα κλικ στο Enable captive portal το πρώτο πράγμα που επιλέγουμε είναι το interface ή τα interfaces στα οποία θα αφορά. Εμεις θέλουμε Captive Portal μόνο για τους wireless clients που συνδέονται πάνω στο WAP, οπότε διαλέγουμε το OPT1. Αφιερώστε λίγο χρόνο και δείτε όλες τις επιλογές. Δεν σκοπεύουμε να βάζουμε τους clients να κάνουν login ή κάτι παρόμοιο, οπότε στην ενότητα Authentication τσεκάρουμε το No Authentication. Αναλόγως του συνολικού διαθέσιμου bandwidth, δεν είναι άσχημη ιδέα να περιορίσουμε την ταχύτητα πρόσβασης κάθε client. Εντάξει, είπαμε να μοιράζουμε δωρεάν Ίντερνετ, αλλά αυτό δεν σημαίνει ότι μπορεί να έρχεται κάποιος με τα torrents του και να καταλαμβάνει όλη τη σύνδεση για εκείνον :) Δώστε προσοχή στη θυρίδα Portal page contents, στο κάτω μέρος της σελίδας. Από εκεί μπορούμε να ανεβάσουμε ένα αρχείο HTML, ώστε κάθε φορά που κάποιος wireless client επιχειρεί να συνδεθεί στο WAP να βλέπει μια συγκεκριμένη ιστοσελίδα. Μέσω αυτής θα μπορούσαμε να απευθύνουμε έναν σύντομο χαιρετισμό στο φίλο ή στη φίλη που θέλει πρόσβαση στο δίκτυό μας, ενδεχομένως και μια παράγραφο με τους όρους χρήσης της υπηρεσίας που παρέχουμε. Α, όλα κι όλα, σ' εμάς αρέσουν οι καθαρές δουλειές! Σε συνδυασμό με το περιεχόμενο της θυρίδας After authentication Redirection URL, αφού ο επισκέπτης κάνει κλικ στο σχετικό κουμπάκι της πρώτης σελίδας, μπο-



*Το pfSense, πάνω στο οποίο είναι συνδεδεμένο το wireless access point του καλού γείτονα, προσφέρει δυνατότητες packet capturing ή sniffing, αν προτιμάτε. Εδώ βλέπουμε ότι τρέχει ήδη ένα packet capture στο WAN interface του pfSense, επομένως καταγράφεται *και* η δικτυακή κίνηση των καλών γειτόνων. Ο διαχειριστής του pfSense δεν έχει θέσει κάποιον περιορισμό στο μέγεθος του αρχείου με τα δεδομένα που σνιφάρονται. Όταν κρίνει πως έχει αρκετά, θα σταματήσει το capturing χειροκίνητα. Μετά, με την ησυχία του θα προχωρήσει σε ένα ωραιότατο packet analysis (βλ. άρθρο στη σελ. 22).*

ρούμε να τον ανακετευθύνουμε σε όποιο site θέλουμε. Κάπως έτσι μπορεί να γίνεται η διαφήμιση, που λέγαμε προηγουμένως. Βεβαίως, από εκεί και πέρα ο επισκέπτης θα μπορεί να σερφάρει όπου θέλει και να χρησιμοποιεί τις υπηρεσίες που επιθυμεί.

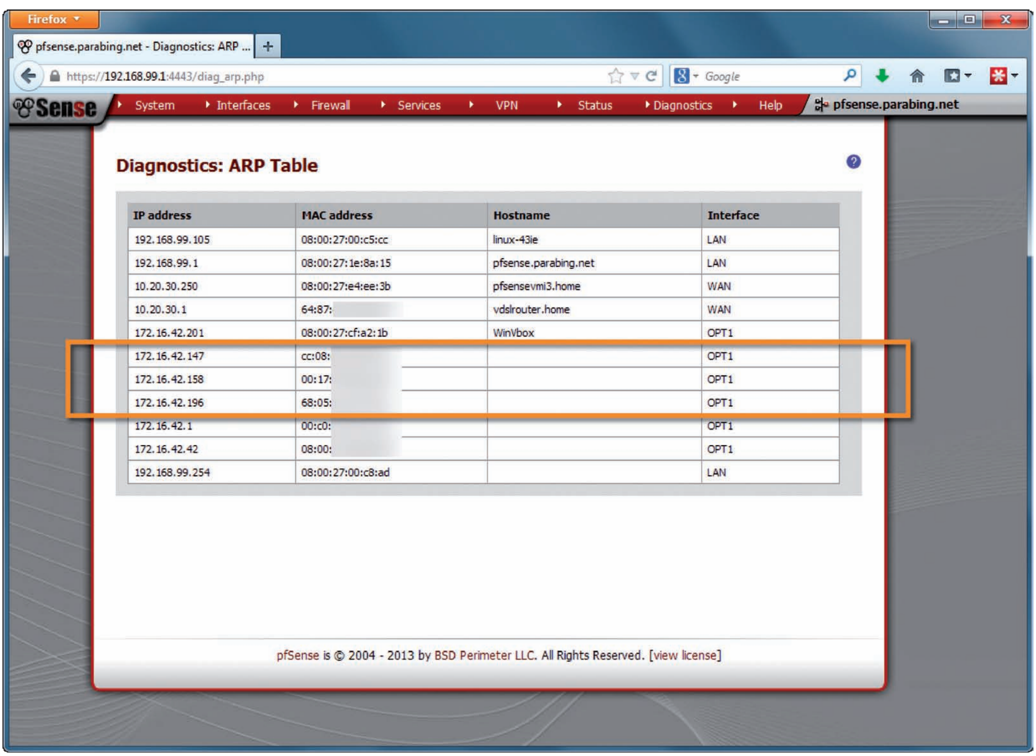
Παρατηρήστε εδώ ότι προκειμένου ν' αποκτήσει ένας client του WAP πρόσβαση στο Ίντερνετ όταν το Captive Portal είναι ενεργοποιημένο για το αντίστοιχο interface του pfSense, θα πρέπει ν' ανοίξει πρώτα κάποιον web browser και να επιχειρήσει να επισκεφτεί έναν οποιονδήποτε δικτυακό τόπο. Θα δει τότε τη σελίδα που έχουμε ανεβάσει στο Captive Portal κι *αφού* κάνει κλικ στο σχετικό κουμπάκι, τότε και μόνον τότε θα αρθούν οι σχετικοί περιορισμοί του firewall και θα μπορεί να βγει στο Ίντερνετ.

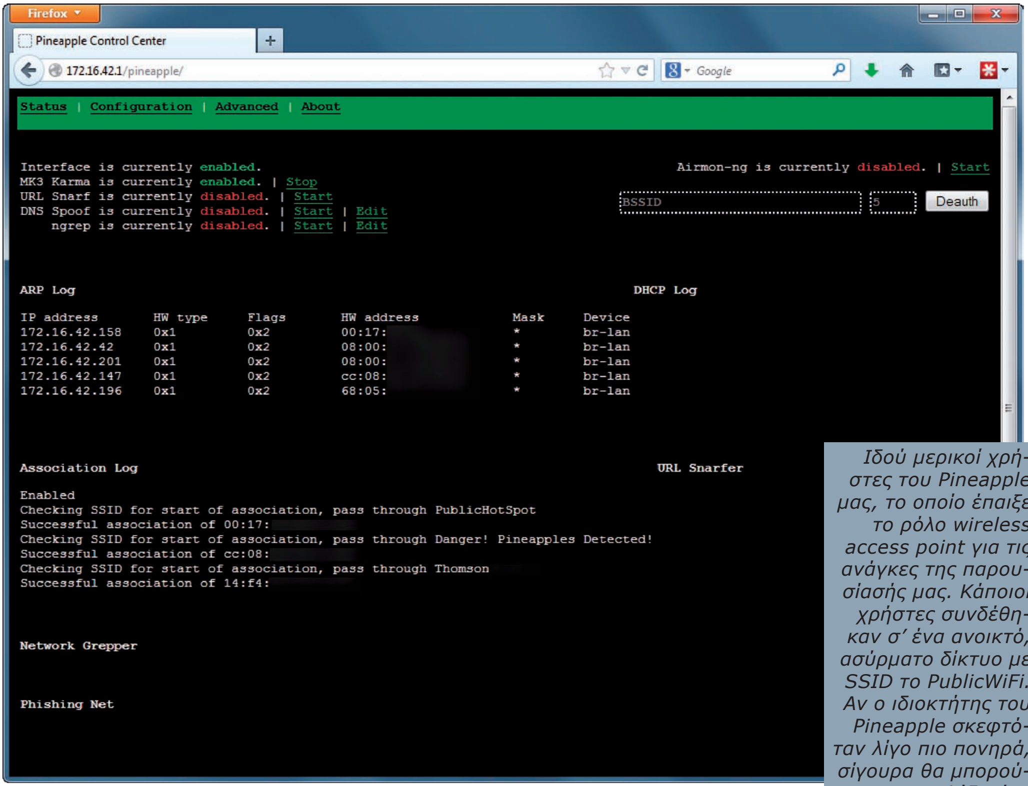
Όσο για εκείνη τη σελίδα που θ' αντικρίζει όποτε επιχειρεί σύνδεση, πέρα από το απλό κείμενο είναι δυνατό να περιλαμβάνει και κάποιες εικόνες. Σε μια τέτοια περίπτωση, αυτές θα τις έχουμε ανεβάσει από τη καρτέλα File Manager (βλ. δεξιά, στη σελίδα του pfSense με τις ρυθμίσεις για το Captive Portal). Αν επιθυμείτε οι χρήστες σας να παίρνουν σοβαρά αυτή την πρώτη σελίδα που θα βλέπουν και πράγματι να χρησιμοποιούν το WAP που με τόση φροντίδα τους έχετε φτιάξει, προτείνουμε να αφιερώσετε και τον ανάλογο χρόνο για τη δημιουργία της υπό συζήτηση σελίδας. Μ' άλλα λόγια, μην κάνετε ό,τι κάναμε εμείς :)

Βελτίωση της εμπειρίας, για τους αγαπητούς καλεσμένους

Από τη στιγμή που το ασύρματο δίκτυο είναι ανοικτό, αργά ή γρήγορα θα το μάθουν στη γειτονιά και κατά πάσα πιθανότητα θα το χρησιμοποιούν

Από το web panel του pfSense βλέπουμε, μεταξύ άλλων, τις διευθύνσεις IP και MAC που έχουν οι clients που χρησιμοποιούν το ανοικτό wireless access point.



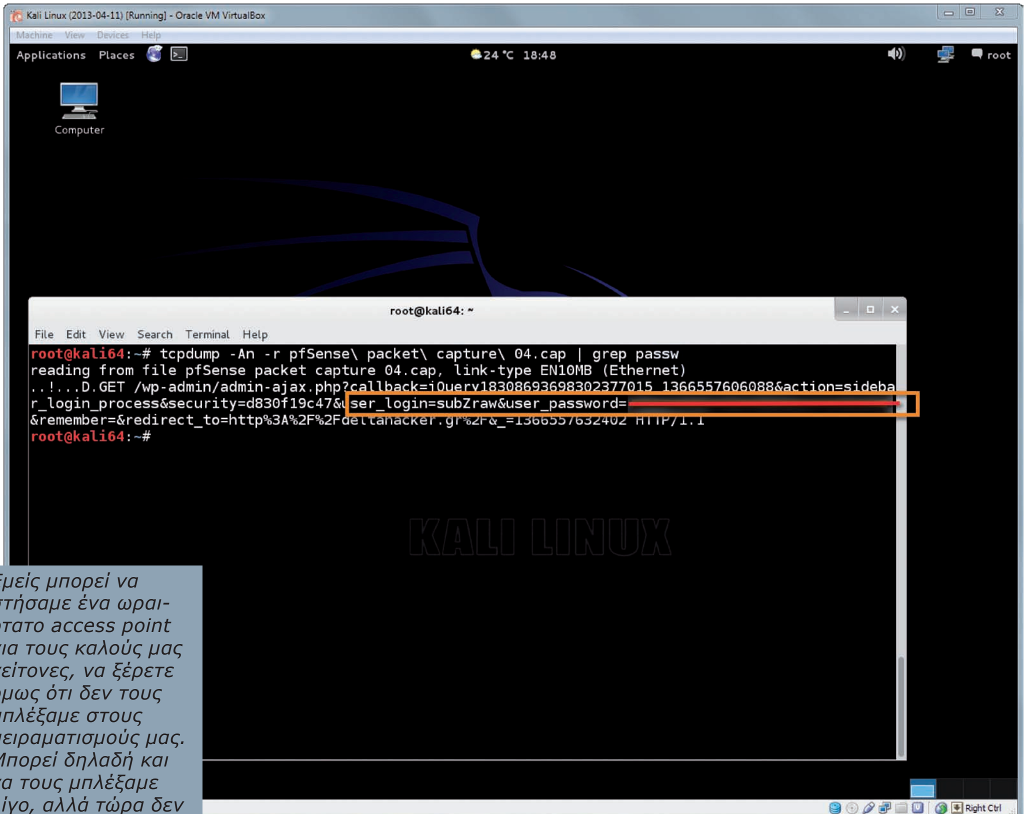


Ιδού μερικοί χρήστες του Pineapple μας, το οποίο έπαιξε το ρόλο wireless access point για τις ανάγκες της παρουσίασής μας. Κάποιοι χρήστες συνδέθηκαν σ' ένα ανοικτό, ασύρματο δίκτυο με SSID το PublicWiFi. Αν ο ιδιοκτήτης του Pineapple σκεφτόταν λίγο πιο πονηρά, σίγουρα θα μπορούσε να επιλέξει ένα όνομα που θα προκαλούσε/προσέλκυε πολύ περισσότερο. Παρατηρήστε ότι οι συσκευές άλλων γειτόνων έσπευσαν να συνδεθούν σ' ένα ανοικτό ασύρματο δίκτυο στο οποίο είχαν συνδεθεί παλιότερα και ακόμα θυμόντουσαν. Τα κατάφεραν μια χαρά — τέτοιο είναι το Pineapple :) Σημειώστε ότι για τις δοκιμές μας χρησιμοποιήσαμε το προηγούμενο μοντέλο του Pineapple, το Mark III, το οποίο έχει πάψει να διατίθεται. Τα χαρακτηριστικά του νεότερου μοντέλου, του Mark IV, μπορείτε να δείτε στο http://wifipineapple.com.

τακτικά. Αναλόγως των επιπτώσεων στο bandwidth που θα διαπιστώσετε, φανταζόμαστε πως θα αποφασίσετε αν έχει νόημα να συνεχίσετε αυτή την παροχή ή όχι. Πάντως, για όσο χρόνο επιτρέπετε την πρόσβαση, είναι καλή ιδέα να έχετε στο pfSense ενεργοποιημένο κάποιο proxy cache — κι αναφερόμαστε φυσικά στο δημοφιλέστατο Squid. Υπάρχουν sites που λίγο πολύ όλοι μας επισκεπτόμαστε, σχεδόν καθημερινά. Οι χρήστες του ανοικτού WAP είναι κι εκείνοι μέρος αυτού του «όλοι μας». Το Squid στο επίπεδο του pfSense, θα αποθηκεύει δημοφιλή στοιχεία που λίγο πολύ όλοι οι web browsers ζητούν. Έτσι, τα οφέλη από την οικονομία bandwidth θα αφορούν τόσο τους χρήστες του WAP, όσο και σ' εμάς.

Για να εγκαταστήσετε το Squid συνδεθείτε στο web panel του pfSense, επιλέξτε System → Packages και κάντε κλικ στην καρτέλα Available Packages. Στη λίστα με τα διαθέσιμα προς εγκατάσταση πακέτα, θα δείτε ότι υπάρχει η «παλιά» αλλά δοκιμασμένη και σταθερή έκδοση του Squid (μιλάμε για τη 2.7.9), καθώς και η νεότερη, της σειράς 3.0, η οποία προς το παρόν είναι σε φάση beta. Έχουμε δοκιμάσει και τις δύο εκδόσεις, χωρίς προβλήματα. Αν μας ρωτούσατε θα προτείναμε τη νεότερη, αν μη τι άλλο επειδή παρέχει δυνατότητα ζωντανής παρακολούθησης των logs του Squid μέσα από το web panel του pfSense. Για την εγκατάσταση, αρκεί ένα κλικ στο εικονίδιο με το συν, στα δεξιά.

Αφού ολοκληρωθεί η εγκατάσταση, πηγαίνετε στο Services → Proxy Server. Αρχικά, υποδείξτε τα network interfaces στα οποία θα αφορά το Squid. Πέρα από το OPT1 που εξυπακούεται, ίσως θελήσετε να βάλε-



Εμείς μπορεί να στήσαμε ένα ωραιότατο access point για τους καλούς μας γείτονες, να ξέρετε όμως ότι δεν τους μπλέξαμε στους πειραματισμούς μας. Μπορεί δηλαδή και να τους μπλέξαμε λίγο, αλλά τώρα δεν το θυμόμαστε. Αν πάντως κάναμε κάτι, ήταν κατά λάθος. Στο screenshot φαίνεται μια πρώτη απόπειρα για ανάλυση ενός packet capture file, το οποίο πήραμε αφού ζητήσαμε από το pfSense να σνιφάρει για λίγο τη δικτυακή κίνηση στο WAN interface. Διακρίνονται τα user name και password του γράφοντα, στο account του στο http://deltahacker.gr. Εντάξει, το password δεν διακρίνεται, αφού η σκέψη να το αποκαλύψουμε και μετά να το αλλάξουμε αμέσως, ε, ομολογούμε ότι μας φόβισε λίγο :S Γι' αυτό και το μουτζουρώσαμε καμιά δεκαριά φορές!

τε το Squid να διαμεσολαβεί και για την κίνηση που ξεκινά από το LAN interface. Φροντίστε να είναι τσεκαρισμένο το Allow users on interface, ώστε να επιτρέπεται η χρήση του proxy από όλους όσοι συνδέονται μέσω των interfaces που επιλέξατε μόλις. Τσεκαρισμένο πρέπει να 'ναι και το Transparent proxy, ώστε να γίνεται χρήση του Squid χωρίς να δηλώνεται στους web browsers των clients. Για να βλέπουν το Captive Portal κι όσοι clients έχουν δηλωμένη ρητά τη διεύθυνση του proxy στον web browser, φροντίστε να είναι ενεργοποιημένο και το Patch captive portal. Τυχόντα μηνύματα που θα εμφανίζει το Squid θα θέλατε να είναι σε κάποια γλώσσα που οι χρήστες κι εσείς θα καταλαβαίνετε, οπότε φροντίστε γι' αυτό από το drop-down menu ονόματι Language (πλαίσιο Logging Settings, λίγο παρακάτω). Στο κάτω μέρος της σελίδας του pfSense, κάντε ένα κλικ στο Save. Οι επιλογές θα ληφθούν υπόψη και το Squid θα ξεκινήσει (ή θα κάνει ένα reload, αν είναι ήδη ενεργό).

Με αυτές τις βασικές επιλογές που μόλις κάνατε, πέρα από δωρεάν πρόσβαση στο Ίντερνετ θα προσφέρετε πλέον στους αγαπητούς γείτονες κι έναν ωραιότατο proxy cache. Έτσι, για να σερφάρουν λίγο γρηγορότερα οι άνθρωποι. Μια πρόσθετη βασική επιλογή που αξίζει να δείτε, είναι στην καρτέλα Local Cache και συγκεκριμένα στο Hard disk cache size. Πιθανώς να θέλετε ν' αυξήσετε τον χώρο που δεσμεύει το Squid από τον δίσκο του pfSense, προκειμένου ν' αποθηκεύει τα στοιχεία που φέρνει από το Ίντερνετ. Εμείς, π.χ., δώσαμε στο Squid ένα gigabyte.

Γενική παρακολούθηση της κίνησης

Ο διαχειριστής ενός WAP βρίσκεται στην προνομιακή θέση να βλέπει και να κάνει πράγματα που οι χρήστες του δεν μπορούν. Πηγαίνετε, για παράδειγμα, στο Services → Proxy server και μετά στην καρτέλα Real time. Από εκεί, σε πραγματικό χρόνο παρακολουθείτε ποιες σελίδες επισκέπτονται και γενικά σε ποιους servers συνδέονται οι clients που χρησιμοποιούν τον proxy. Κι επειδή ο τελευταίος είναι transparent, οι πληροφορίες που παρακολουθείτε αφορούν σε όλους τους χρήστες του WAP.

Θέλετε ακόμη πιο λεπτομερείς πληροφορίες για την κίνηση που διέρχεται από το pfSense; Κανένα πρόβλημα, απλά εγκαταστήστε το πακέτο ntop. Από το web panel του pfSense, δώστε System → Packages και στη λίστα της καρτέλας Available Packages εντοπίστε το ntop. Εγκαταστήστε το μ' ένα κλικ στο κουμπάκι με το συν, στα δεξιά της γραμμής. Μετά πηγαίνετε στο Diagnostics → ntop Settings. Ορίστε ένα password για τον διαχειριστή της εφαρμογής και στο πλαίσιο με τα διαθέσιμα interfaces επιλέξτε εκείνα που θέλετε να παρακολουθεί το ntop. Αν διαλέξετε το WAN, τότε το ntop θα ασχολείται με την κίνηση του pfSense ανεξαρτήτως αν προέρχεται από χρήστες του WAP ή από χρήστες του LAN, πίσω από τον router. Επίσης, το ntop θα εξετάζει και την κίνηση που φτάνει στο WAN interface από τον έξω κόσμο. Για ν' αποκτήσετε άμεσα πρόσβαση στο web interface του ntop κι έτσι να δείτε τι μπορεί να καταγράφει, κάντε ένα κλικ στο Access ntop. Γενικά, μετά την πρώτη ρύθμιση μπορείτε να μεταβαίνετε στο ntop επιλέγοντας Diagnostics → ntop, κατευθείαν από το web panel του pfSense.

Αυτή είναι η σελίδα του απλού Captive Portal που δημιουργήσαμε. Εσείς μην κάνετε ό,τι (δεν) κάναμε και φτιάξτε μια πολύ πιο όμορφη κι ελκυστική — για να μην πούμε καθησυχαστική :) Στη δική μας, πάντως, θέλαμε να προειδοποιείται με σαφήνεια ο γείτονας, για το πού πάει να μπλέξει. Παρατηρήστε ότι μετά το κλικ στο κουμπάκι, ο χρήστης του access point κατευθύνεται αυτόματα στο http://deltahacker.gr. Ε, να μας κάνουμε και λίγη διαφήμιση! Από εκεί και μετά, πάντως, ο χρήστης είναι ελεύθερος να επισκεφτεί όποιο site θέλει.



Συνήθειες ενός περίεργου οικοδεσπότη

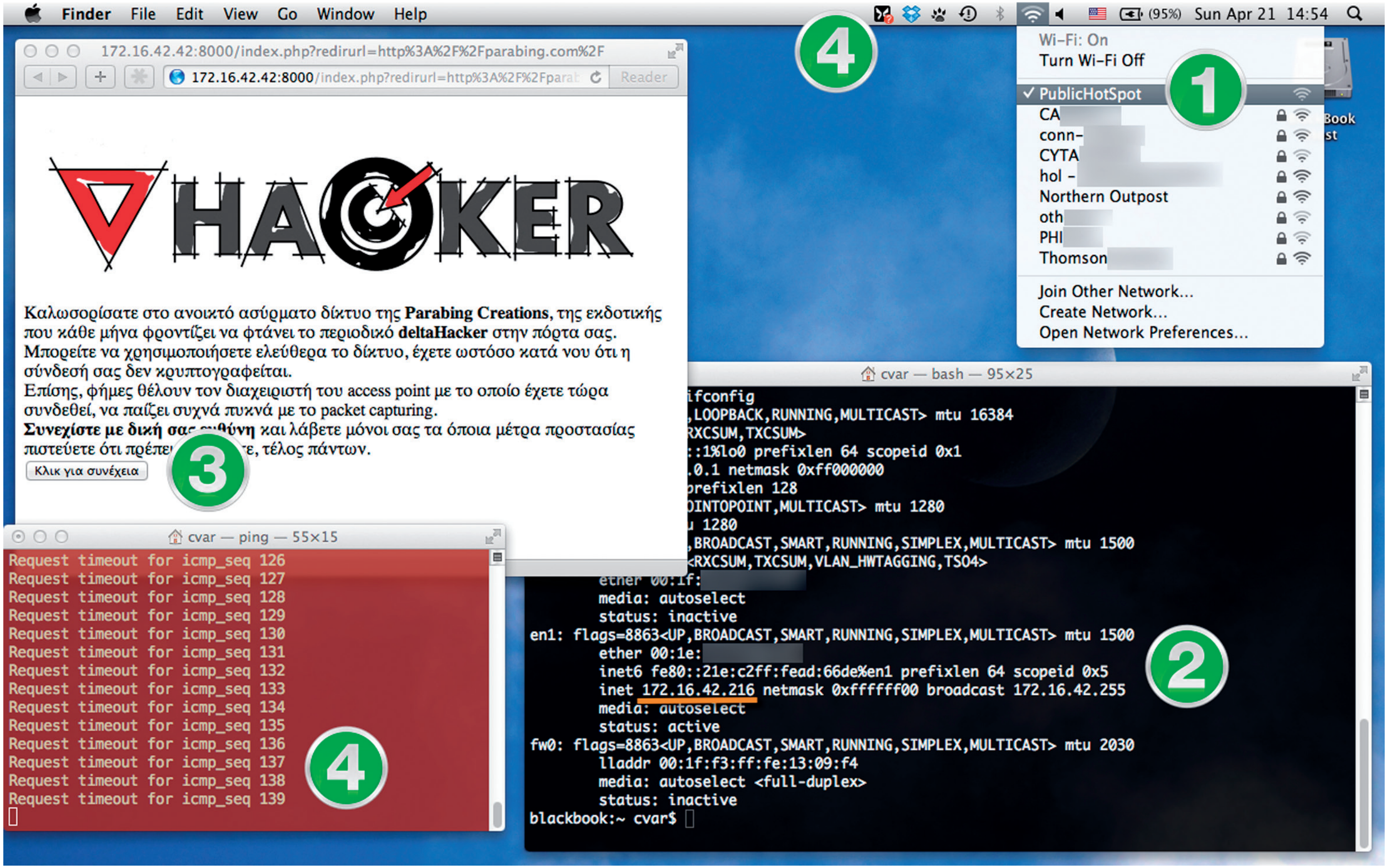
Το να σνιφάρουμε τα πακέτα που διακινούνται στο δικό μας δίκτυο, είναι σίγουρα δικαίωμά μας. Όταν όμως αυτό το δίκτυο το χρησιμοποιούν κι άλλοι, έστω κι εμμέσως, τότε εγείρονται κάποια ηθικά ζητήματα — για να μην πούμε και νομικά. Αν βέβαια οι χρήστες του δικτύου έχουν συναινέσει σε μια πολιτική που επιτρέπει, μεταξύ άλλων, και το packet sniffing ή capturing, τότε τα πράγματα αλλάζουν κι ως διαχειριστές μπορούμε να δρούμε ήσυχοι κι ανενόχλητοι.

Με την προϋπόθεση λοιπόν πως έχουμε εξηγήσει με σαφήνεια στους χρήστες του WAP ότι **α)** η δικτυακή τους κίνηση δεν κρυπτογραφείται και **β)** ο διαχειριστής έχει απεριόριστη ελευθερία ενεργειών, τότε μπορούμε επιτέλους να εφαρμόσουμε κι εδώ μερικές από τις τεχνικές packet analysis που περιγράφονται στο άρθρο που ξεκινά από τη σελίδα 22. Βεβαίως, πρώτα πρέπει να συλλέξουμε μερικά πακέτα για ανάλυση. Από το web panel του pfSense, αρκεί να δώσουμε Diagnostics → Packet Capture. Στη σελίδα που θα δούμε, αρχικά ορίζουμε το interface από το οποίο θα καταγράφουμε πακέτα. Προφανώς, αν βάλουμε το WAN τότε θα καταγράφουμε τα πάντα. Στις άλλες επιλογές, το μόνο που εμείς κάνουμε είναι να θέτουμε Count = 0, ώστε το packet capturing να μη σταματά μόνο του αλλά να συνεχίζει έως ότου αποφασίσουμε εμείς να το σταματήσουμε. Πρέπει πάντως να είστε προσεκτικοί μ' αυτή την επιλογή, αφού αναλόγως των πακέτων που διακινούνται από το pfSense το αρχείο του packet capture είναι πιθανό να μεγαλώσει, γρήγορα. Για όση ώρα διαρκεί το packet capturing δεν χρειάζεται να έχουμε ανοικτή τη σχετική σελίδα του pfSense. Όταν αποφασίσουμε πως έχουμε αρκετά πακέτα για ανάλυση, ερχόμαστε ξανά στο Diagnostics → Packet Capture και κάνουμε κλικ στο Stop. Αμέσως μετά το pfSense θα παραθέσει τα περιεχόμενα του packet capture file σε ένα πλαίσιο, στο κάτω μέρος της σελίδας Diagnostics: Packet Capture. Το αρχείο μπορούμε να το κατεβάσουμε για περαιτέρω ανάλυση με ένα κλικ στο κουμπάκι Download Capture. Αν το εν λόγω αρχείο είναι μεγάλο, ο web browser μας θα αποτύχει να αποδώσει σωστά τη σελίδα και θα δούμε μόνο το απέραντο ...λευκό. Σε μια τέτοια περίπτωση αρκεί ένα refresh και θα εμφανιστεί ξανά η σελίδα, χωρίς το πλαίσιο αλλά με το κουμπάκι Download Capture στη θέση του. Άπαξ λοιπόν και πάρουμε το αρχείο μπορούμε να ξεκινήσουμε την ανάλυσή του. Για άλλη μια φορά, συμβουλευτείτε το άρθρο της σελίδας 22.

Μέτρα προστασίας

Ασχέτως αν συνδεόμαστε στο ανοικτό WAP του καλού γείτονα, στο δημόσιο HotSpot της πλατείας ή σ' εκείνο της καφετέριας, εννοείται ότι ποτέ δεν στέλνουμε ευαίσθητες πληφορορίες μέσω μη κρυπτογραφημένων καναλιών. Το ίδιο φυσικά ισχύει και για ασύρματες συνδέσεις που γίνονται από HotSpots σε ξενοδοχεία, αεροδρόμια, πλοία κ.ο.κ. Το γεγονός ότι καλωσοριζόμαστε από μια όμορφη, επαγγελματικά σχεδιασμένη σελίδα του αντίστοιχου Captive Portal, δεν σημαίνει ότι παρασκηνιακά δεν λαμβάνει χώρα ένα ανελέητο packet capturing. Αν μη τι άλλο, μην ξεχνάτε τον παράγοντα «βαριεστημένος διαχειριστής με άπειρο ελεύθερο χρόνο».

Εκτός λοιπόν από το casual web browsing, όπως το λένε στα χωριά αρκετών εξ ημών, όταν πρόκειται να κάνουμε login σε κάποιο site όπου διατηρούμε λογαριασμό και με τίποτε δεν θέλουμε ν' αποκτήσουν κι άλλοι πρόσβαση σ' αυτόν, βεβαιωνόμαστε ότι η σύνδεση είναι HTTPS.



Αφού συνδεθεί κάποιος σε ένα wireless access point με Captive Portal, ουσιαστικά η πρόσβασή του στο Ίντερνετ είναι μπλοκαρισμένη έως ότου κάνει login, πληκτρολογήσει κάποιον κωδικό/αναγνωριστικό ή/και αποδεχθεί τους όρους χρήσης της υπηρεσίας. Στο εικονιζόμενο παράδειγμα, ο χρήστης του access point με όνομα PublicHotSpot (1) έχει πάρει διεύθυνση IP (2) και ανοίγοντας τον web browser βλέπει την απλή σελίδα του Captive Portal (3). Πριν κάνει κλικ στο κουμπί, πριν δηλαδή αποδεχθεί τους όρους χρήσης της υπηρεσίας, παρατηρήστε ότι είναι μπλοκαρισμένος (4).

Σε περίπτωση που πάρουμε προειδοποίηση για μη έγκυρο πιστοποιητικό, δεν συνεχίζουμε. Ακόμη καλύτερα, όσον αφορά στην εγκυρότητα των πιστοποιητικών καλό είναι να χρησιμοποιούμε κάποιο add-on σαν το Certificate Patrol (`http://patrol.psyced.org`).

Ανάλογες παρατηρήσεις εξάλλου ισχύουν και για άλλα πρωτόκολλα. Όταν, π.χ., πρόκειται να κάνετε μια μεταφορά μέσω FTP, προτιμήστε συνδέσεις FTP πάνω από SSH. Γενικότερα, ένας τρόπος για να σερφάρετε έχοντας το κεφάλι σας ήσυχο είναι μέσω ενός κρυπτογραφημένου, ασφαλούς καναλιού SSH. Δείτε το σχετικό άρθρο στο Parabing για το τι εννοούμε:

`http://parabing.com/?p=552`

Ακόμη καλύτερα –και για να μην προστατεύετε μόνο τη δικτυακή κίνηση που αφορά στον εκάστοτε web browser, φροντίστε ώστε να βγαίνετε στο Ίντερνετ μέσω OpenVPN. Στο τεύχος 007 του deltaHacker θα βρείτε άρθρο με αναλυτικές πληροφορίες για το πώς θα στήσετε τον δικό σας OpenVPN server σε περιβάλλον Ubuntu. Στο site μας υπάρχει και αντίστοιχο άρθρο για το στήσιμο του OpenVPN σε περιβάλλον FreeBSD:

`http://deltahacker.gr/?p=5669`

Παρατηρήστε τώρα ότι ο υποψιασμένος διαχειριστής του WAP που σκοπεύει να κάνει δύσκολη τη ζωή των χρηστών του —καμία σχέση μ’ εμάς, δηλαδή— θα λάβει μέτρα ώστε να δυσκολέψει τις συνδέσεις SSH, καθώς κι εκείνες του OpenVPN. Στην περίπτωση του παραδείγματός μας, όπου έχουμε να κάνουμε με το pfSense, το μόνο που χρειάζεται είναι μερικοί κανόνες στο firewall για το μπλοκάρισμα των συνδέσεων στα ports 22/TCP, 1194/TCP και 1194/UDP: Το port 22/TCP είναι το προκαθορισμένο για το SSH, ενώ τα 1194/TCP και 1194/UDP είναι τα προκαθορισμένα ports για το OpenVPN. (Μπορείτε να διαβάσετε πολλά περισσότερα για τα ports και τη λογική τους στο άρθρο που ξεκινά από τη σελίδα 24 του deltaHacker 018.)

Τι γίνεται σ’ αυτή την περίπτωση; Πολύ απλά, έχουμε φροντίσει από πριν ώστε ο SSH server ή/και ο OpenVPN server που συνδεόμαστε ν’ ακούει για αιτήσεις από μη-τυποποιημένα, αναπάντεχα ports. Έτσι, ο διαχειριστής του WAP είναι μάλλον απίθανο να ’χει προβλέψει ότι ο OpenVPN server μας χρησιμοποιεί το port 50129/UDP, οπότε μάλλον δεν θα ’χει και κάποιον σχετικό, απαγορευτικό κανόνα στο firewall. Αν βέβαια διαθέτει άπειρο χρόνο, μετά από προσεκτική ανάλυση ενός packet capture file θα δει ότι κάτι συμβαίνει μ’ αυτό το 50129/UDP, οπότε ίσως προσθέσει έναν καινούργιο κανόνα στο firewall. Αλλά κοιτάξτε όμως τι έγινε: Αντί να μας κάνει τη ζωή δύσκολη, τελικά δυσκολέψαμε εμείς τη δική του. Και στην τελική, ακόμη κι όταν καταλάβει τι γίνεται κι ενημερώσει σχετικά το firewall, το πιθανότερο είναι ότι εμείς θα ’χουμε ήδη χάσει κάθε ενδιαφέρον προς το WAP του — ή απλά θ’ αλλάξουμε το port του server.

Καλή διασκέδαση με τα wireless access points και τα Captive Portals — απ’ όποια μεριά του firewall κι αν βρίσκεστε!

deltahacker.gr